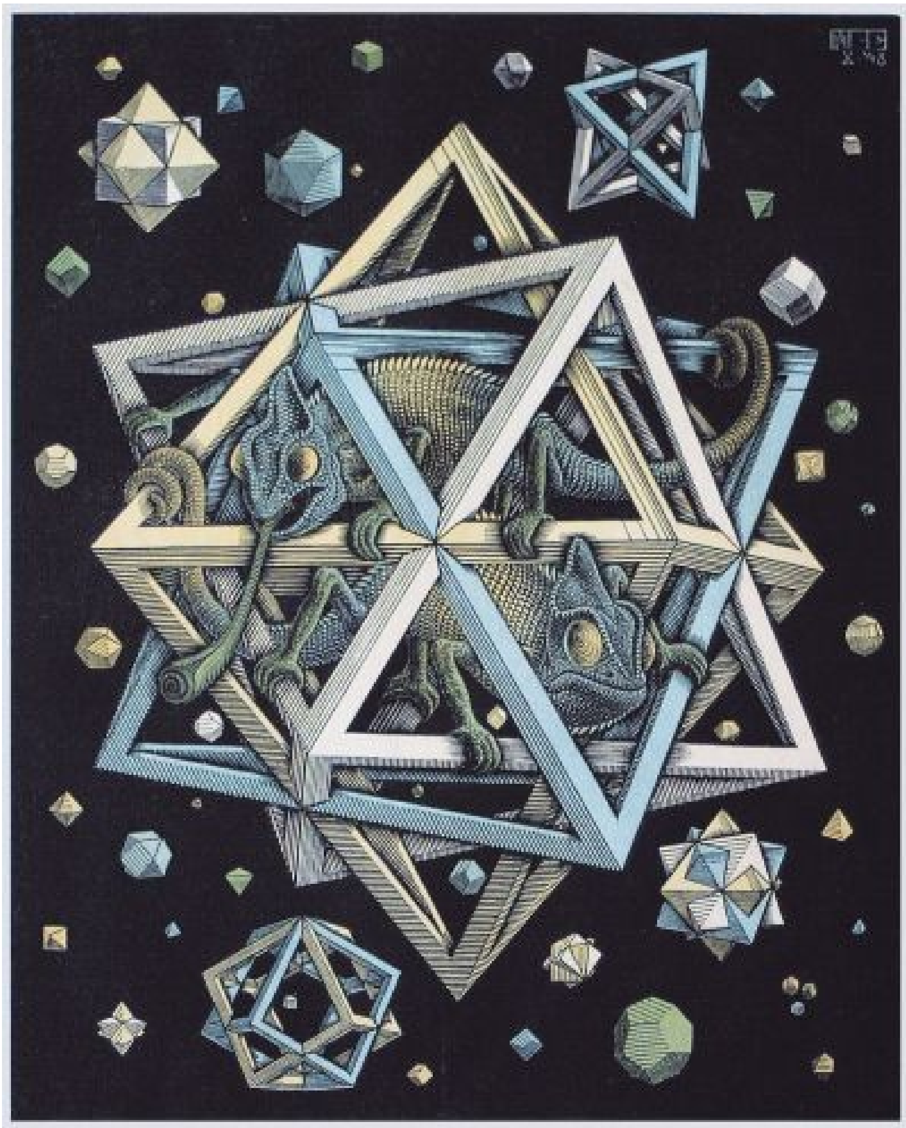
Many-Sorted Logic

From scratch up to

Gödel's Completeness Theorem



by A. B. Zeidler

Many-Sorted Logic

A. Bernhard Zeidler

思考

Contents

1 What do we have here?

Overview

In this text we introduce many-sorted languages as a more natural way to formalize mathematics than the common one-sorted logic. While this is not really necessary from a rigid point of view (all mathematics can be covered with set-theory, which can be done in the one-sorted language of sets) there are many good reasons to use many-sorted languages instead:

- Almost all mathematics speaks of several sorts of objects: Points and lines in plane geometry, vectors and scalars in linear algebra and even sets and classes in formal set theory. It stands to reason to formalize this, as it is.

- Many-sorted languages are no more complicated than one-sorted languages, and it only takes little longer to formulate this machinery.

- While it is possible to boil down many-sorted languages with finitely many sorts into a one-sorted language and use predicates (i.e. a 1-ary relation symbol saying *I am of sort "i"*) it is harder to prove that this construction preserves all the theorems (like Gödel's Completeness Theorem) than to start from scratch.

So what we try to do here is to present an introduction into many-sorted languages and models and to formulate and prove the most important theorems of formal logic, like the correctness, completeness and compactness theorems. We will also present examples of common many-sorted languages along the way, including the language of modules over commutative rings, of ring-extensions and that of sets and classes. If you are not familiar with naive mathematical logic yet, read remark 34 first.

My approach to formal languages follows the extremely elegant and efficient path my esteemed teacher Prof. Ulrich Felgner of the University of Tübingen laid out for one-sorted languages. Part of my thesis required a generalization to many-sorted languages. As he never wrote a textbook on this himself, I try to preserve part of his marvelous insights herein.

Introduction

David Hilbert gave a set of axioms for Euclidean Geometry, i.e. the geometry of points, lines and planes in 3-dimensional space. His first axiom reads as: "For any two points there exists a straight line passing through them". In more formal terms this would read as

$$\forall\, P \text{ point } \;\forall\, Q \text{ point } \;\exists\, g \text{ line } : \; (P \in g \text{ and } Q \in g)$$

So let us take a closer look at this: We have 3 sorts of objects: points, lines and planes. Lines and planes consist of points, hence there is a relation, e.g. $P \in g$ that designates whether a point P is contained in the line g. Note that this relation is between points and lines. There is another such relation between points and planes. They are usually denoted by the same symbol $\in$ but from a formal point of view these are two distinct relations.

This becomes even more important, if we regard the intersection $\cap$ of lines. It could be formalized as a function that turns two lines into a point (with $\emptyset$ being a special point that we could pick up as a constant). Likewise the intersection of planes yields a line (with another $\emptyset$ being a special line). We even have an intersection of a line and a plane, which should be a point. That is we need to distinguish between three separate functions:

$$\begin{array}{rcccc} \cap_L & : & \{\,\text{lines}\,\} \times \{\,\text{lines}\,\} & \to & \{\,\text{points}\,\} \\ \cap_P & : & \{\,\text{planes}\,\} \times \{\,\text{planes}\,\} & \to & \{\,\text{lines}\,\} \\ \cap_T & : & \{\,\text{lines}\,\} \times \{\,\text{planes}\,\} & \to & \{\,\text{points}\,\} \end{array}$$

What we find here is that functions start from (one or more) objects of predefined sort(s) and yield another object of another, fixed sort. Like relations you cannot enter just any object of any sort into a function.
As always the quantifiers $\forall$ and $\exists$ translate into *for all* and *there is* but we note, that they always apply to one sort of objects only, e.g. for all points P there is this and that. Hence every sort has its own quantifiers, too.

All these considerations are in no way artificial - this is what mathematicians do all the time: Thinking about different types of objects and their interaction. But before we venture on to giving a formal definition of the structure of many-sorted languages let us present another example from a different field: Linear Algebra. In the theory of vector-spaces (or modules over a ring) we have a base field F and a vector-space V. The base field features the operations of addition and multiplication. The vector-space only has an addition, but there also is a scalar multiplication between scalars and vectors. Altogether we have 4 binary functions:

$$\begin{array}{rcccc} +_f & : & F \times F & \to & F \\ \cdot_f & : & F \times F & \to & F \\ +_v & : & V \times V & \to & V \\ \cdot_s & : & F \times V & \to & V \end{array}$$

Note that we also have constants, the neutral elements 0_f and 1_f in F and 0_v in V. One of the properties required for a vector-space is, that for any two scalars a, $b \in F$ and for any vector $x \in V$ we have $(a+b)x = (ax) + (bx)$.
Note that in this equation the first addition is $+_f$ and the second is $+_v$. But this only is the usual operator-overloading commonplace in mathematics. What we want to stress here, is the use of quantifiers: What has been covered in $a \in F$ or $x \in V$ will turn into different quantifiers in a many-sorted language. So this formula would read, as:

$$\forall_f\, a \;\; \forall_f\, b \;\; \forall_v\, x \;:\; (a +_f b) \cdot_s x \;=\; (a \cdot_s x) +_v (b \cdot_s x)$$

It might be a little cumbersome to track precisely which $+$ belongs to which sort, if you are familiar with these expressions there is no loss in omitting the indices. But this is not the case, when it comes to quantifiers! It is safe to rewrite $\forall_f\, a$ as $\forall\, a \in F$ but it is important to note, that it only applies to all objects of the sort F. So in the end you will read the following formula (next page) in a textbook about linear algebra, but remember - this is natively a formula of a two-sorted language!

$$\forall a \in F \;\; \forall b \in F \;\; \forall x \in V \; : \; (a+b)x \; = \; (ax) + (bx)$$

From these two examples it should be apparent, that the natural way to formalize mathematics is a structure that allows us to speak of different sorts of objects. Every sort has its own quantifiers and constants. Relations and functions can link objects of different sorts but are sort-dependent, as well: You cannot insert a scalar into a slot where a vector is supposed to be.

Back to Where it all Began

If you, dear reader, are still in doubt what all that fuzz is about, why we need to further formalize mathematics anyway, let us relay Russel's Paradox that crashed Cantor's notion of sets: Let us take a look at the set of all sets, that do not contain themselves, formally $\Omega := \{ X \mid X \notin X \}$. As of Cantor's notion this is a totally legit construction: It is a set of objects X that satisfy a reasonable condition, namely $X \notin X$.

So the question is: Is $\Omega \in \Omega$? If "yes" then we would have $\Omega \notin \Omega$, by the definition of Ω, a contradiction. If "no" this is $\Omega \notin \Omega$ and hence we should have had $\Omega \in \Omega$, another contradiction. Either way the coin lands, we arrive at a contradiction - so what is wrong?

Alas, the answer is not a short one: Cantor's notion is just too far reaching to be free of such contradictions. What we need to do is what has been called *reduction of size*. That is we have to start with very simple sets, $\emptyset$ for example, and then slowly work ourselves up to more complex sets. Given two (or more) sets the constructions allowed are intersections, unions and power sets. These then turn up other nice sets. We can even build up Cartesian products of sets this way. So whatever we can build up using these constructions is safe. So mathematicians build up the naturals, integers, rationals, reals, complex numbers and so on from scratch - only using $\emptyset$ and these constructions. All this is done in the formal language of sets. This is a one-sorted language, all the constants and variables are considered sets.

However it is tempting to talk about the *set of all sets* - this essentially is what Ω has been in the first place. Or at least the set of all groups. But it can be shown that any (non-empty) set can be equipped with a group structure, so this does not help either.

But category theory talks about such concepts all the time - they even have functors, that map an object from one category to another (e.g. we start with a commutative ring R and end up with the topological space $\mathrm{Spec}(R)$). So how do they avoid these paradoxa? Simple: By avoiding the word *set* at the wrong times.

It is perfectly fine to speak about the *class* of all sets. The problem was, that Cantor allowed a language that could speak of itself. If Ω is a class (not of the sort *set*) then the relation $\Omega \in \Omega$ is forbidden. So in modern set and category theory there are two sorts: sets and classes. Some classes are small enough to be sets themselves, but not necessarily so.

There are two ways to tackle this from a formal point of view: First of all we can take the variables to be classes and introduce a predicate for classes saying *I also am a set* or we could use a two-sorted language, with one sort being sets, the other being classes. Both approaches have their charms.

2 Defining Many-Sorted Languages

Let us start by giving a definition of formal languages with several sorts. Admittedly the definition is somewhat lengthy but nevertheless understandable at first glance. The simplicity of the definition is due to a peculiarity that it has in common with for example Cantor's "definition" of sets: The definition is (more or less) informal and appeals to the readers common sense, not some advanced mathematical knowledge. In case you begin to wonder what all this is about, or need an example of what we are aiming at, consult the next section 3 or take a look at example 7.

Of course there is some danger to this - as we have pointed out Cantor's definition allowed to construct Russel's Paradox. In our case, however, things are free of hazards, as we only give rules that can be formalized easily on the basis of naive set theory. Yet we wish to omit a precise formalization, as this wouldn't make anything clearer, but only obfuscate the simplicity of the formalism.

The first object we require are the words over a given alphabet. Formal languages will then be defined by fixing a certain alphabet and giving a collection of rules that determine whether any given word belongs to the language or not. Intuitively speaking let $\mathcal{A}$ be any collection of symbols satisfying the following two properties:

- any two different symbols of $\mathcal{A}$ look different and
- no symbol of $\mathcal{A}$ is part of another symbol of $\mathcal{A}$

Then such a collection is said to be an **alphabet**. If we write some (i.e. finitely many) symbols of $\mathcal{A}$ in a row then this sequel of symbols is said to be a **word** over $\mathcal{A}$. The collection of all words over $\mathcal{A}$ is denoted by $\mathcal{A}^*$. More formally

$$a_1 a_2 \dots a_k \in \mathcal{A}^* \quad :\Longleftrightarrow \quad a_1 \in \mathcal{A}, \dots, a_k \in \mathcal{A}$$

Nota that the above two properties guarantee that any word can be read unambiguously. If for example we had two symbols $a = \cdot$ and $b = \cdot\cdot$ then the word $\cdot\cdot\cdot$ could be read as ab, ba or even aaa. Yet this may not occur due to the second property. The idea of words over a given alphabet should be clear enough now, but here's a rigid definition based on naive set theory:

Definition 1:
Let $\mathcal{A}$ be any non-empty set, that we use as a collection of symbols. Then the set $\mathcal{A}^*$ of words over $\mathcal{A}$ is be defined to be

$$\mathcal{A}^* \; := \; \bigcup_{k \in \mathbb{N}} \mathcal{A}^k$$

That is a word $a = a_1 a_2 \dots a_k$ is just an n-tuple $a = (a_1, a_2, \dots, a_k) \in \mathcal{A}^k$ for some non-negative integer $k \in \mathbb{N}$. The case $k = 0$ includes the **empty word** which is denoted by ε. The number k with $a \in \mathcal{A}^k$ is called the **length** $\ell(a)$ of the word, denoted by

$$\ell(a_1 \dots a_k) \; := \; k$$

The **concatenation** ab of words $a = a_1a_2 \dots a_m$ and $b = b_1b_2 \dots b_n \in \mathcal{A}^*$ is just the word $ab = a_1a_2 \dots a_m b_1b_2 \dots b_n \in \mathcal{A}^*$. Doing this $\mathcal{A}^*$ becomes a monoid under the concatenation, $\ell : \mathcal{A}^* \to \mathbb{N}$ is a homomorphism of such. If $b \in \mathcal{A}^*$ is **part of** $d \in \mathcal{A}^*$ we write $b \mid d$, formally that is

$$b \mid d \quad :\Longleftrightarrow \quad \exists\, a, c \in \mathcal{A}^* \;:\; abc = d$$

And if we wish to refer to the i-th symbol of the word a (where $i \in 1 \dots \ell(a)$ is any number from 1 to the length of a) we write $a[i]$, formally

$$a[i] \;:=\; a_i \;\in\; \mathcal{A} \quad \text{for} \quad a \;=\; a_1 \dots a_k \;\in\; \mathcal{A}^*$$

Example 2:
The Latin alphabet consists of $2 \cdot 26$ symbols, namely the common letters

$$\mathcal{A} \;:=\; \{\, a, b, c, \dots, z, A, B, C, \dots, Z \,\}$$

Thus $word \in \mathcal{A}^*$ is a word over the Latin alphabet and the first sentence of the fourth gospel of John

In the beginning, before the creation of the world, he was, who is the word.

is a sequel of 15 words separated by interpunctional signs. Note however that these signs (space, comma and full stop) do not even belong to the alphabet. If the Latin alphabet was appended by these three symbols, then the whole sentence would be a single word over this appended alphabet.

Definition 3:
A many-sorted language $\mathcal{L}$ is based on four ingredients: A collection $\mathcal{C}$ of constant symbols, a collection $\mathcal{F}$ of function symbols and a collection $\mathcal{R}$ of relation symbols. Hereby we require that all these symbols can be distinguished from one and another. Further any symbol has a certain signature of sorts as described below. The collection of all sorts available is thereby denoted by I. Then we can define what we mean by a (many-sorted, first order) formal language $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ in several steps:

- **The symbols:** The alphabet $\mathcal{A}$ of the language $\mathcal{L}$ consists of the following symbols:
  - a **left bracket** (and a **right bracket**)
  - two logical symbols: the **negation** $\neg$ and **implication** $\to$
  - for every sort $i \in I$ a symbol of **equality** $=_i$, a **universal quantifier** $\forall_i$ and an **existential quantifier** $\exists_i$
  - for each **sort** $i \in I$ and every natural number $j \in \mathbb{N}$ we fix a **variable symbol** $x_{i,j}$ and this is assigned $\text{sort}(x_{i,j}) := i \in I$.
  - every **constant symbol** $c \in \mathcal{C}$ given, where c is supposed to already have been assigned a certain $\text{sort}(c) \in I$.
  - every **function symbol** $f \in \mathcal{F}$ given, where f is supposed to already have been assigned a signature of the form $\text{sort}(f) \in I^{k+1}$ for some $1 \leq k \in \mathbb{N}$.

– every **relation symbol** $R \in \mathcal{R}$ given, where R is supposed to already have been assigned a signature of the form $\text{sort}(R) \in I^k$ for some $1 \leq k \in \mathbb{N}$.

We will now define what we mean by a *term* or *formula* of the language $\mathcal{L}$. I.e. we decree that some words over the above alphabet will be called terms, others will be called formulae. However we will not give an explicit definition, but instead give rules how such words may be obtained. Then a term (or formula respectively) is simply a word which was created using the rules given. Further any term will be assigned a sort in I simultaneously to its construction.

- **The terms:** We call a word $t \in \mathcal{A}^*$ a **term** of the language $\mathcal{L}$, iff it can be generated by applying the following rules finitely many times:

  (T1) any variable symbol x of $\mathcal{L}$ already is a term of $\mathcal{L}$, and this term is assigned the sort $\text{sort}(x)$.

  (T2) any constant symbol c of $\mathcal{L}$ already is a term of $\mathcal{L}$, and this term is assigned the sort $\text{sort}(c)$.

  (T3) for any function symbol f of $\mathcal{L}$ with $\text{sort}(f) = (i_1, \dots, i_k, i_{k+1})$ and for any (previously generated) terms $t_1, \dots, t_k \in \text{term}(\mathcal{L})$ of $\mathcal{L}$, having the sorts $\text{sort}(t_1) = i_1, \dots, \text{sort}(t_k) = i_k$ we decree that the word $f t_1 \dots t_k \in \mathcal{A}^*$ is a term of $\mathcal{L}$ of $\text{sort}(f t_1 \dots t_k) := i_{k+1}$.

- **Atomic formulae:** We call a word $\varphi \in \mathcal{A}^*$ an **atomic formula** of the language $\mathcal{L}$, iff it is of first or second type:

  (A1) for any two terms s, t of the same sort $i := \text{sort}(s) = \text{sort}(t)$, the word $s =_i t \in \mathcal{A}^*$ is an atomic formula of $\mathcal{L}$.

  (A2) for any relation symbol R of the sort $\text{sort}(R) = (i_1, \dots, i_k)$ and any terms $t_1, \dots, t_k \in \text{term}(\mathcal{L})$ of $\text{sort}(t_1) = i_1$ up to $\text{sort}(t_k) = i_k$ the word $R t_1 \dots t_k \in \mathcal{A}^*$ is an atomic formula of $\mathcal{L}$.

- **Formulae:** We call a word $\varphi \in \mathcal{A}^*$ a **formula** of the language $\mathcal{L}$, iff it can be generated by applying the following rules finitely many times:

  (F1) any atomic formula φ of $\mathcal{L}$ already is a formula of $\mathcal{L}$.

  (F2) for any two previously generated formulae φ and ψ of $\mathcal{L}$, both the words $(\neg \varphi) \in \mathcal{A}^*$ and $(\varphi \to \psi) \in \mathcal{A}^*$ are formulae of $\mathcal{L}$, too.

  (F3) for any variable symbol x of $\mathcal{L}$ of the sort $i := \text{sort}(x)$ and any previously generated formula φ of $\mathcal{L}$, both the words $(\exists_i \, x \, \varphi) \in \mathcal{A}^*$ and $(\forall_i \, x \, \varphi) \in \mathcal{A}^*$ are formulae of $\mathcal{L}$, too.

Let us denote the set $\text{var}(\mathcal{L}) := \{ x_{i,j} \mid i \in I, j \in \mathbb{N} \}$ of variable symbols and the set $\text{quant}(\mathcal{L}) := \{ \exists_i \mid i \in I \} \cup \{ \forall_i \mid i \in I \}$ of quantifier symbols of $\mathcal{L}$. The set of all terms of $\mathcal{L}$ will be denoted by $\text{term}(\mathcal{L})$, the set of atomic formulae of $\mathcal{L}$ by $\text{atom}(\mathcal{L})$ and - not surprisingly - the set of all formulae of $\mathcal{L}$ by $\text{form}(\mathcal{L})$. By rule (F1) we have $\text{atom}(\mathcal{L}) \subseteq \text{form}(\mathcal{L})$. Now the set $\mathcal{L}$ itself is finally defined by

$$\mathcal{L} \quad := \quad \text{term}(\mathcal{L}) \cup \text{form}(\mathcal{L}) \quad \subseteq \quad \mathcal{A}^*$$

This definition only provides the minimum basis of the mathematical formalism, of course. It is presented in this way to ensure unique readability, but is not yet perfectly suited to be truly used in mathematics. Thus in formal logic the formulae regarded have a tendency to become very long and intricate. In order to remain on top of things we need some simplifications to put them into a form easier to read and hence introduce the following notational conventions:

Notation 4:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a many-sorted language, then we introduce the following notational conventions concerning terms and formulae

(i) If I consists of one element only $I = \{i\}$, we say that $\mathcal{L}$ is a one-sorted language. In this case we don't need the sort function and can omit all the indexes i on variable symbols $x_{i,j}$, equality relations $=_i$ and the quantifiers $\forall_i$ and $\exists_i$.

(ii) By a theorem of Post any junctor can be expressed using a combination of $\neg$ and $\rightarrow$. Therefore we can use any junctor as an expression of $\mathcal{L}$, defining its usage by a logically equivalent formula composed of $\neg$ and $\rightarrow$. In this text we will frequently employ the abbreviations

$$\begin{aligned} \varphi \vee \psi \quad &:\Longleftrightarrow \quad (\neg\varphi) \rightarrow \psi \\ \varphi \wedge \psi \quad &:\Longleftrightarrow \quad \neg(\varphi \rightarrow (\neg\psi)) \\ \varphi \leftrightarrow \psi \quad &:\Longleftrightarrow \quad (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi) \end{aligned}$$

Nota we distinguish between single-stroke arrows, as in $\varphi \rightarrow \psi$ that are part of the formal language and double-stroke arrows, as in (x is a cat) $\Longrightarrow$ (x is a mammal), which belong to the meta-language. As in the example of the implication here, we will build our formalism of deduction in such a way, that $\rightarrow$ will behave just like $\Longrightarrow$.

(iii) We follow the usual conventions concerning the bracketing of terms and formulae, i.e. in order to save some brackets, we decree that $\neg$ is of higher priority than $\wedge$ and $\vee$, which in turn are considered of higher priority than $\rightarrow$ and $\leftrightarrow$. As an example the formula

$$\neg\varphi \vee \psi \ \rightarrow \ \chi \wedge \omega$$

is a shorter notation for the (more obfuscated but formally correct) formula

$$(((\neg\varphi) \vee \psi) \ \rightarrow \ (\chi \wedge \omega))$$

(iv) We apply brackets to functions, i.e. we write $f(t_1, \dots, t_k)$ instead of $ft_1 \dots t_k$. The same is true for relations - we write $R(t_1, \dots, t_k)$ instead of $Rt_1 \dots t_k$. An exception to this are binary relations which are usually written in the form sRt instead of Rst. (Nota that the comma here doesn't even belong to the symbols of the formal language).

Regard the (nonsensical) formula $\varphi := \forall\, y\,(x = y \vee \exists\, x\,(x = z))$ then φ contains the variable symbols x, y and z. Yet there is a difference: The variable symbol y is quantified over and hence cannot be "seen from the outside". Hence y is called a *bound* variable of φ, whereas x and z are said to be *free* variables of φ. We will emphasize this by writing $\varphi(x, z)$ when needed.
But this is not the sole difference, in the sub-formula $\exists\, x\,(x = z)$ the variable symbol x is bound and hence only the first occurence of x in φ is free, the second is bound. Hence if we want to *substitute* the variable x by a term t this is only allowed in the first instance. Hence the substitution of x by t in φ turns out to be $\varphi[x : t] = \forall\, y\,(t = y \vee \exists\, x\,(x = z))$. Let us take a general look at these notions in the following:

Definition 5:
Let again $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a many sorted language, then we introduce the following notions for terms and formulae of $\mathcal{L}$. As these were defined recursively it stands to reason that we may revert this recursion and decompose terms and formulae into their atomic parts. The following definitions now make use of this concept

(i) **Sub-formulae:** For any formula $\varphi \in \text{form}(\mathcal{L})$ define the set $\text{sub}(\varphi) \subseteq \text{form}(\mathcal{L})$ of all **sub-formulae** of φ (by inverse recursion) to be the following

$$\begin{aligned}
\text{sub}(\varphi) &:= \{\varphi\} \text{ if } \varphi \text{ is atomic} \\
\text{sub}(\neg\varphi) &:= \{(\neg\varphi)\} \cup \text{sub}(\varphi) \\
\text{sub}(\varphi \to \psi) &:= \{(\varphi \to \psi)\} \cup \text{sub}(\varphi) \cup \text{sub}(\psi) \\
\text{sub}(\exists_i\, x\, \varphi) &:= \{(\exists_i\, x\, \varphi)\} \cup \text{sub}(\varphi) \\
\text{sub}(\forall_i\, x\, \varphi) &:= \{(\forall_i\, x\, \varphi)\} \cup \text{sub}(\varphi)
\end{aligned}$$

For any two formulae $\varphi, \psi \in \text{form}(\mathcal{L})$ we say that φ is a *sub-formula* of ψ, which we will abbreviate by $\varphi \le \psi$, iff $\varphi \in \text{sub}(\psi)$ is a subformula of ψ.

(ii) **Free variables:** Let $a \in \mathcal{L}$ be a term or formula of $\mathcal{L}$, then we define the set of **free variables** of a (by inverse recursion again) to be the following

$$\begin{aligned}
\text{free}(x) &:= \{x\} \\
\text{free}(c) &:= \emptyset \\
\text{free}(f t_1 \dots t_k) &:= \text{free}(t_1) \cup \dots \cup \text{free}(t_k) \\
\text{free}(R t_1 \dots t_k) &:= \text{free}(t_1) \cup \dots \cup \text{free}(t_k) \\
\text{free}(s =_i t) &:= \text{free}(s) \cup \text{free}(t) \\
\text{free}(\neg\varphi) &:= \text{free}(\varphi) \\
\text{free}(\varphi \to \psi) &:= \text{free}(\varphi) \cup \text{free}(\psi) \\
\text{free}(\exists_i\, x\, \varphi) &:= \text{free}(\varphi) \setminus \{x\} \\
\text{free}(\forall_i\, x\, \varphi) &:= \text{free}(\varphi) \setminus \{x\}
\end{aligned}$$

where we used the same notations as in definition 3 and the sorts of all terms are supposed to be adequate. To express that φ is a formula having the free variables $\mathrm{free}(\varphi) = \{ x_1, \ldots, x_k \}$ we will occasionally write $\varphi(x_1, \ldots, x_k)$. Analogously we will write $\varphi(x_1, \ldots, x_k, \ldots)$ in the case that $x_1, \ldots, x_k$ are some (but not necessarily all) free variables of φ, i.e. in case $\{ x_1, \ldots, x_k \} \subseteq \mathrm{free}(\varphi)$.

(iii) **Sentences:** A formula φ of $\mathcal{L}$ is called to be a **sentence**, iff it does not contain any free variables and we denote the set of all such by

$$\mathrm{sen}(\mathcal{L}) \;:=\; \{ \varphi \in \mathrm{form}(\mathcal{L}) \mid \mathrm{free}(\varphi) = \emptyset \}$$

(iv) **Free occurence:** If (for $k \in 1 \ldots \ell(\varphi)$) $\varphi[k] \in \mathrm{var}(\mathcal{L})$ is a variable symbol $x := \varphi[k]$ then we say that this is a **bound occurence** of x in φ, iff at this place x is quantified over. That is $\varphi = \alpha\varphi'\beta$ where (1) $\varphi' \leq \varphi$ is a subformula of the form $\varphi' = Qx\,\varphi''$ for some quantifier $Q \in \mathrm{quant}(\mathcal{L})$ and (2) the location $\varphi[k]$ is part of φ''. If $\varphi[k] \in \mathrm{var}(\mathcal{L})$ is a variable symbol $x = \varphi[k]$, that is *not* quantified over, we say this is a **free occurence** of x in φ.

(v) **Freely substitutable:** Let $x \in \mathrm{var}(\mathcal{L})$ be a variable symbol, $t \in \mathrm{term}(\mathcal{L})$ be a term and $\varphi \in \mathrm{form}(\mathcal{L})$ be a formula of $\mathcal{L}$. Then we say, that x is **freely substitutable** by t, iff $\mathrm{sort}(x) = \mathrm{sort}(t)$ and at any free occurence of x in φ the variable x is not part of some sub-formula of the form $Qy\,\varphi'' \leq \varphi$ where $Q \in \mathrm{quant}(\mathcal{L})$ is quantifier and $y \in \mathrm{free}(t)$ also is a free variable of t. Formally:

$$\mathrm{free}(t) \cap \left\{ y \in \mathrm{var}(\mathcal{L}) \;\middle|\; \begin{array}{ll} \varphi' \leq \varphi & \text{sub-formula} \\ \text{such that} & \varphi' = Qy\,\varphi'' \\ \text{and} & x \in \mathrm{free}(\varphi') \end{array} \right\} \;=\; \emptyset$$

(vi) **Substitution:** Let $x \in \mathrm{var}(\mathcal{L})$ be a variable symbol and $t \in \mathrm{term}(\mathcal{L})$ be a term of the same sort, $\mathrm{sort}(x) = \mathrm{sort}(t)$, and $a \in \mathcal{L}$ be a term or formula. We will now define the notion of **substituting** x by t in a using inverse recursion. Thus we start with $\ell(a) = 1$, in this case we let

$$a[x:t] := \begin{cases} a & \text{for } a \neq x \\ t & \text{for } a = x \end{cases}$$

If now $2 \leq \ell(a)$, $Q \in \mathrm{quant}(\mathcal{L})$ is any quantifier and $y \in \mathrm{var}(\mathcal{L})$ is any variable symbol of $\mathcal{L}$, then we recursively define the substitution $a[x:t]$ of x by t in a, by letting

$$a[x:t] := \begin{cases} f t_1[x:t] \ldots t_k[x:t] & \text{for } a = f t_1 \ldots t_k \\ R t_1[x:t] \ldots t_k[x:t] & \text{for } a = R t_1 \ldots t_k \\ (t_1[x:t] =_i t_2[x:t]) & \text{for } a = (t_1 =_i t_2) \\ (\neg\varphi[x:t]) & \text{for } a = (\neg\varphi) \\ (\varphi_1[x:t] \to \varphi_2[x:t]) & \text{for } a = (\varphi_1 \to \varphi_2) \\ Q\,y\,\varphi[x:t] & \text{for } a = Q\,y\,\varphi \text{ and } x \neq y \\ Q\,y\,\varphi & \text{for } a = Q\,y\,\varphi \text{ and } x = y \end{cases}$$

Example 6:
So what is this *freely substitutable* business all about? Why care? As a first example let us take a look at the formula $\varphi = \exists\, y\, (y = x)$. It has the free variable x only. So what happens if we replace x with a third variable z?

$$\varphi[x : z] \quad = \quad \exists\, y\, \big(y = z\big)$$

Nothing actually, the logical content of $\varphi[x : z]$ is precisely the same, both formulae express a truism, as we could take $y = x$ or $y = z$ respectively. Now we replace x by the term $t = y + z$, in this case we get

$$\varphi[x : (x + y)] \quad = \quad \exists\, y\, \big(y = y + z\big)$$

In presence of the group axioms this would be $\exists\, y(0 = z)$ So the truth of the statement suddenly depends on the content of z. For $z = 0$ it would be true, but false in any other case. So what happened? We replaced x in a position where the formula quantifies over the variable y, that also occurs in the term t by which we substituted. This is precisely the situation that we exclude by demanding that x shall be freely substitutable by t.

As a second example consider $\varphi = \forall\, u\, ((u = v + x) \wedge \forall\, v\, (v = u + x))$. Let us first mark any free appearance of the variables u, v and x in the formula φ by placing a dot on top of the respective symbol

$$\varphi \quad = \quad \forall\, u\, \big((u = \dot{v} + \dot{x}) \wedge \forall\, v\, (v = u + \dot{x})\big)$$

- In φ the free variable x *is not* freely substitutable by the term $t := u$, as $\varphi' := (u = v + x) \wedge \forall\, v\, (v = u + x)$ is a sub-formula, with free variable $x \neq u$ such that $\varphi = \forall\, u\, \varphi'$ and u is a variable of t.
- In φ the free variable x *is not* freely substitutable by the term $t := v + x$, as $\varphi' := (v = u + x)$ is a sub-formula with free variable $x \neq v$ such that $\forall\, v\, \varphi' \leq \varphi$ and v is a variable of t.
- In φ the free variable x *is* freely substitutable by the term $t := x + x$ as the quantifiers in φ do not meet the variable x of the term t.
- In φ the free variable v *is* freely substitutable by the term $t := x$ as in the *one* free appearance of v in φ the quantifier $\forall\, u$ does not bind a variable of t. In fact, by definition, the substitution leaves the right part of the formula unchanged

$$\varphi[v : x] \quad = \quad \forall\, u\, \big((u = x + x) \wedge \forall\, v\, (v = u + x)\big)$$

3 Examples of Many-Sorted Languages

Example 7:
Let us begin with a neat little example: plane geometry. Here we have two sorts of objects - points and lines - which we represent by two sorts $I = \{ p, \ell \}$. If the point P lies on the line L we would like to write $P \in L$. For lines we have the operation of the intersection $\cap$ and the relation $\|$ of being parallel. Altogether that is

$$\begin{aligned} \mathcal{C} &= \emptyset \\ \mathcal{F} &= \{ \cap \} \\ \mathcal{R} &= \{ \in, \| \} \end{aligned}$$

The sorts are clear: First of all $\|$ is a relation between lines, therefore sort($\|$) $= (\ell, \ell)$. Now $P \in L$ shall indicate that P lies on L, hence $\in$ has to have the sort($\in$) $= (p, \ell)$. And as $\cap$ turns two lines into a point we have sort($\cap$) $= (\ell, \ell, p)$. The following formula expresses, that *any two lines, that are not parallel, intersect in a point*:

$$\forall_\ell K \, \forall_\ell L \, \Big(\neg (K \| L) \to \exists_p P \, \big(K \cap L =_p P \big) \Big)$$

Nota we already used the standard notation $K \cap P := \cap(K, L)$ here to increase the readability of the formula. And we will continue to do so in the following examples with all binary operations like $+$, $-$, $\cdot$ and :.

If we want to add a far point, as in projective geometry, we can do so as a constant symbol, say $\mathcal{C} = \{ \infty \}$ of the sort(∞) $= p$. Using this we could formulate, that *any two distinct, parallel lines intersect in* ∞:

$$\forall_\ell K \, \forall_\ell L \, \Big(\big((K \| L) \wedge \neg (K =_\ell L) \big) \to (K \cap L =_p \infty) \Big)$$

Example 8:
Let us continue with a one-sorted example: The language $\mathcal{L}_r$ of rings. To be one-sorted means $I = \{ r \}$ contains a single element only. So what do we need for rings? The constants 0 and 1, the addition $+$ and multiplication $\cdot$ are a minimum. So we start with

$$\begin{aligned} \mathcal{C} &= \{ 0_r, 1_r \} \\ \mathcal{F} &= \{ +_r, \cdot_r \} \\ \mathcal{R} &= \emptyset \end{aligned}$$

As we only have one sort r the constants are of this sort of course sort(0_r) $= r$ and sort(1_r) $= r$. And as $+$ and $\cdot$ are binary operations we find their sort to be sort($+_r$) $= (r, r, r)$ and sort($\cdot_r$) $= (r, r, r)$. We keep using the index r, as we will build upon this example, later on. If we want to prove a formula in this language, for any ring, we need to start with the properties of rings. These are sentences in this language. First of all a ring is a commutative group under $+$ with neutral element 0, that is

$$\text{(G1)} \quad \forall_r a \, \forall_r b \, \forall_r c \quad a +_r (b +_r c) \; =_r \; (a +_r b) +_r c$$

$$
\begin{array}{lll}
\text{(G2)} & \forall_r a\, \forall_r b & a +_r b \;=_r\; b +_r a \\
\text{(G3)} & \forall_r a & a +_r 0_r \;=_r\; a \\
\text{(G4)} & \forall_r a\, \exists_r b & a +_r b \;=_r\; 0_r
\end{array}
$$

The next set of formulae expresses that a ring is a commutative monoid under $\cdot$ with neutral element 1 and that $+$ and $\cdot$ are interlocked by the law of distributivity. If (G1) to (G4) and (R1) to (R4) are satisfied, we have a *commutative ring* (with unity)

$$
\begin{array}{lll}
\text{(R1)} & \forall_r a\, \forall_r b\, \forall_r c & a \cdot_r (b \cdot_r c) \;=_r\; (a \cdot_r b) \cdot_r c \\
\text{(R2)} & \forall_r a\, \forall_r b & a \cdot_r b \;=_r\; b \cdot_r a \\
\text{(R3)} & \forall_r a & a \cdot_r 1_r \;=_r\; a \\
\text{(R4)} & \forall_r a\, \forall_r b\, \forall_r c & a \cdot_r (b +_r c) \;=_r\; (a \cdot_r b) +_r (a \cdot_r c)
\end{array}
$$

As the addition $+$ has an inverse, the subtraction $-$ it can be helpful to also introduce this to our language, that is we even take $\mathcal{F} = \{+,-,\cdot\}$ where $\mathrm{sort}(-) = (r,r,r)$, as well. We will call this expanded language the *language of rings* and denote it by

$$
\mathcal{L}_r \;:=\; \mathcal{L}_{\{r\}}\Big(\{0_r, 1_r\}, \{+_r, -_r, \cdot_r\}, \emptyset, \mathrm{sort}\Big)
$$

Picking up this function symbol we need to fix the meaning of $-$ as the inverse of $+$. So in this case we need another property, namely

$$
\text{(AI)} \quad \forall_r a\, \forall_r b\, \forall_r c \quad (c -_r b =_r a) \;\leftrightarrow\; (c =_r a +_r b)
$$

We will later use the *theory of integral* domains. This is the set T_{id} of sentences (G1) to (G4), (R1), $\ldots$, (R4), (AI) and (ID) where

$$
\text{(ID)} \quad \forall_r a\, \forall_r b\, \forall_r c \quad (a \cdot_r b =_r 0) \;\rightarrow\; (a =_r 0 \vee b =_r 0)
$$

The only property that is missing for a field is: Any $a \neq 0$ has a multiplicative inverse, which is expressed in the formula $\forall_r a(\neg(a =_r 0_r) \rightarrow \exists_r b\, a \cdot_r b =_r 1_r)$. This gives rise to a division function $:_r$ of $\mathrm{sort}(:_r) = (r,r,r)$ that we include for the *language of fields* (where we rename the sort from r to f)

$$
\mathcal{L}_f \;:=\; \mathcal{L}_{\{f\}}\Big(\{0_f, 1_f\}, \{+_f, -_f, \cdot_f, :_f\}, \emptyset, \mathrm{sort}\Big)
$$

We fix the meaning of : as the inverse of $\cdot$ in the next formula. So for a field we require the properties (G1) to (G4), (R1) to (R4), (AI) and (MI) where

$$
\text{(MI)} \quad \forall_f a\, \forall_f b\, \forall_f c \quad \Big(\neg(a =_f 0_f) \rightarrow \big((c :_f b =_f a) \leftrightarrow (c =_f a \cdot_f b)\big)\Big)
$$

We will consider *algebraically closed* fields later on. In order to express that the variables of $\mathcal{L}_f$ belong to an algebraically closed field we need a whole scheme of axioms: For any $1 \leq k \in \mathbb{N}$ let us abbreviate $z^k := z \cdot_f z \cdot_f \cdots \cdot_f z$ (k-times). Then for each $1 \leq n \in \mathbb{N}$ we pick up the statement

$$
(\mathrm{AC}_n) \quad \forall_f a_0 \,\ldots\, \forall_f a_n\, \exists_f z \quad (a_n \cdot_f z^n) +_f \cdots +_f (a_1 \cdot_f z) +_f a_0 =_f 0_f
$$

Then the *theory of algebraically closed fields* T_{ac} consists of the statements (G1) to (G4), (R1) to (R4), (AI), (MI) and (AC_n) for any $1 \leq n \in \mathbb{N}$.

Example 9:
Our next example addresses a basic notion of linear algebra: Modules over (commutative) rings. Likewise we could append the properties for the base ring to be a field to have the notion of a vector-space. Clearly this language requires two sorts $I = \{ r, m \}$, one for the base ring r, the other for the module m. The ring has the constants 0_r and 1_r, the module 0_m only. There are no relations but the addition $+_r$ and multiplication $\cdot_r$ of scalars, the addition $+_m$ of vectors and a scalar multiplication $\cdot_s$. Altogether this is

$$\begin{aligned} \mathcal{C} &= \{ 0_r, 1_r, 0_m \} \\ \mathcal{F} &= \{ +_r, \cdot_r, +_m, \cdot_s \} \\ \mathcal{R} &= \emptyset \end{aligned}$$

The assignment of sorts is obvious, resp. well-known here, but it has to be given beforehand: $\mathrm{sort}(0_r) = r$, $\mathrm{sort}(1_r) = r$, $\mathrm{sort}(0_m) = m$, continued by $\mathrm{sort}(+_r) = (r, r, r)$, $\mathrm{sort}(\cdot_r) = (r, r, r)$, $\mathrm{sort}(+_m) = (m, m, m)$ and finally $\mathrm{sort}(\cdot_s) = (r, m, m)$. So much for the definition of the language, let's see what we can do with it: We can literally repeat (G1) to (G4) and (R1) to (R4) to express that the variables of sort r belong to a commutative ring. Next the variables of sort m belong to a (left) module over this ring. That is they form a commutative group under $+_m$ with neutral element 0_m. It is easy to repeat these properties in terms of formluae

$$\begin{array}{lcl} \text{(M1)} & \forall_m x\, \forall_m y\, \forall_m z & x +_m (y +_m z) \;=_m\; (x +_m y) +_m z \\ \text{(M2)} & \forall_m x\, \forall_m y & x +_m y \;=_m\; y +_m x \\ \text{(M3)} & \forall_m x & x +_m 0_m \;=_m\; x \\ \text{(M4)} & \forall_m x\, \exists_m y & x +_m y \;=_m\; 0_m \end{array}$$

It takes another 4 formula to express that $\cdot_s$ truly is a scalar multiplication between the variables of sort r and sort m. In fact these are compatibility conditions between all the 4 operations involved

$$\begin{array}{lcl} \text{(S1)} & \forall_r a\, \forall_m x\, \forall_m y & a \cdot_s (x +_m y) \;=_m\; (a \cdot_s x) +_m (a \cdot_s y) \\ \text{(S2)} & \forall_r a\, \forall_r b\, \forall_m x & (a +_r b) \cdot_s x \;=_m\; (a \cdot_s x) +_m (b \cdot_s x) \\ \text{(S3)} & \forall_r a\, \forall_r b\, \forall_m x & (a \cdot_r b) \cdot_s x \;=_m\; a \cdot_s (b \cdot_s x) \\ \text{(S4)} & \forall_m x & 1_r \cdot_s x \;=_m\; x \end{array}$$

Note that all these formulae are sentences - all the free variables have been quantified over. This is commonplace for sets of axioms - what use would it have to employ variables just as variables? Also we see that substituting is a completely natural thing to do, e.g. regard the formula $\varphi(x) = \exists_m z : x +_m z =_m 0_m$, then we might want to replace the variable x by the term $t = (x +_m y)$. Doing this we end up with $\psi(x, y) := \varphi[x : t] = \exists_m z : (x +_m y) + z =_m 0_m$ that now has two free variables, x and y.

Example 10:
In the next example let us introduce the language of ring extensions $S : R$ - this is a language containing two sorts $I = \{ r, s \}$, one for the subring R, the other for the larger ring S. It obviously has the constants and functions (but no relations again)

$$\begin{aligned} \mathcal{C} &= \{ 0_r, 1_r, 0_s, 1_s \} \\ \mathcal{R} &= \emptyset \\ \mathcal{F} &= \{ +_r, \cdot_r, +_s, \cdot_s, \iota \} \end{aligned}$$

The assignment of sorts is the same as in example 9, with the exception of ι, which is a 1-ary function of the sort sort$(\iota) = (r, s)$. We will explain the role of ι below. So much for the language, to express that both R and S are commutative rings, we would have to repeat the respective 2 times 8 formulae (G1) to (R4) of example 8. However then we've just got two (commutative) rings (with identity) and we haven't said a word about the fact that R is contained in S. To do this we included the function symbol ι: The following formulae express that ι truly embeds R into S

$$\begin{array}{lll} \text{(E1)} & \forall_r a \, \forall_r b & \iota(a +_r b) \; =_s \; \iota(a) +_s \iota(b) \\ \text{(E2)} & \forall_r a \, \forall_r b & \iota(a \cdot_r b) \; =_s \; \iota(a) \cdot_s \iota(b) \\ \text{(E3)} & \forall_r a \, \forall_r b & a \; =_r \; b \; \leftrightarrow \; \iota(a) \; =_s \; \iota(b) \\ \text{(E4)} & & \iota(1_r) \; =_s \; 1_s \end{array}$$

This example demonstrates how we can remedy the disjointness of the sorts that this formalism would otherwise yield. We will see this again, in the next example:

Example 11:
As a final example let us present the language of modern set theory. It obviously contains two sorts of variables again $I = \{ s, c \}$ which we interpret as sets and classes respectively. It contains a constant, two relations and a function that takes sets to classes:

$$\begin{aligned} \mathcal{C} &= \{ \emptyset \} \\ \mathcal{R} &= \{ \in_s, \in_c \} \\ \mathcal{F} &= \{ \iota \} \end{aligned}$$

Hereby we assign sort$(\emptyset) = s$, sort$(\in_s) = (s, s)$, sort$(\in_c) = (s, c)$ and as before sort$(\iota) = (s, c)$. Note that by this construction it is already forbidden to speak of a class that is element of another class - sets and classes can only contain sets. The *axiom of extensionality* is a formula that expresses, that any two classes B and C are equal if and only if they share the same sets x as elements:

$$\forall_c B \, \forall_c C \; : \; \big(B =_c C \; \leftrightarrow \; \forall_s x \; (x \in_c B \; \leftrightarrow \; x \in_c C) \big)$$

A *small* class is a class S that also is a set. In terms of the embedding ι this can be expressed elegantly, in the following very simple formula

$$\exists_s X \; : \; S \; =_c \; \iota(X)$$

Nota that many-sorted languages can even mimic the workings of second-order languages in which it is allowed to quantify over relation and function symbols (not only over variable symbols) by introducing a sort for the relation symbols and another for the function symbols. But with the introduction of the language $\mathcal{L}(\in)$ of sets, second-order languages have become next to pointless already, so we will not pursue this path any further.

It is tempting to write out an universal language, that can cover all of algebra, maybe all of mathematics. Well, the language of sets and classes is such an universal language in some sense, but this is highly non-specific. So wouldn't it be nice to have a language for all of algebra, for example? The answer however is: No, it is not!

Mathematics already has an universal language - the metalanguage of naive logic and set theory, refined by their axioms in formalized form. What formal logic provides is another tool how to denote and prove theorems in any other field of mathematics. Instead of prescribing how proofs are supposed to be, we should rather look at formal logic as an asset that opens up alternative ways of proving theorems. It should not be seen as an alternative, but rather as an enhancement.

And to do so we do not need the one universal language, but rather several, lean languages specifically tailored for the problem at hand. Without going into details we want to give an example of what formal logic can accomplish: quantifier elimination.

The first example of quantifier elimination is the determinant. Consider a square matrix A over a field F. It is well known, that A is invertible if and only if its determinant is non-zero. That is we have the equivalent statements

(a) $\exists B : (AB = \mathbb{1}_n) \wedge (BA = \mathbb{1}_n)$

(b) $\neg(\det(A) = 0)$

The advantage is clear: In (a) we would have to check an infinite number of possible inverse matrices B, whereas in (b) we can simply perform a finite computation. So if we can eliminate the quantifiers from a formula there is a finite way to determine its truth. Other examples are the Euclidean algorithm to determine whether a and b have common denominators. Or Buchberger's algorithm to determine whether a polynomial is contained in an ideal. To show off, let us provide some advanced results, without proof [that are already formulated using the notions we are about to introduce in the following sections]:

Definition 12:

(i) Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be any many-sorted, formal language and let $\eta \in \text{form}(\mathcal{L})$ be a formula therein. The we say that η is **quantifier-free**, if it can be generated, as a formula, by rules (F1) and (F2) alone.

(ii) A theory T (that is a set of sentences $T \subseteq \text{sen}(\mathcal{L})$) is said to **admit quantifier-elimination**, iff for any formula $\varphi \in \text{form}(\mathcal{L})$ there is another formula $\eta \in \text{form}(\mathcal{L})$ such that

(1) η is quantifier-free, and

(2) $T \vdash (\varphi \leftrightarrow \eta)$

(iii) If $(X, \varrho) \in \mathrm{real}(\mathcal{L})$ is a realization of $\mathcal{L}$, then we say that (X, ϱ) **admits quantifier-elimination**, iff its theory $\mathrm{th}(X, \varrho)$ admits quantifier elimination, where we define

$$\mathrm{th}(X, \varrho) \;\; := \;\; \{ \varphi \in \mathrm{sen}(\mathcal{L}) \mid (X, \varrho) \models \varphi \}$$

Theorem 13:

(i) Consider the language $\mathcal{L}_f$ of fields introduced in 8, then the theory T_{ac} of algebraically closed fields admits quantifier elimination.

(ii) Consider the language $\mathcal{L}_r$ of rings introduced in 8 and let R be an integral domain (i.e. a realization of $\mathcal{L}_r$ that satisfies $R \models T_{id}$). Then the following statements are equivalent

  (a) R admits quantifier-elimination

  (b) R is finite, or an algebraically closed field

Nota a theory T admits quantifier elimination iff it is *substructure-complete* [3] 3.1. Then it can be shown (introducing the *amalgation property*) [3] 3.2 that T_{ac} is substructure-complete [3] 3.4. An algorithm for quantifier-elimination in algebraically closed fields is presented in [6]. The proof of (ii) is a beautiful synthesis of algebraic arguments supplemented by the benefits of quantifier-elimination [3] 8.11.

4 Formal Deduction

Now we know what is meant by speaking of formulae and these of course shall express mathematical facts (e.g. theorems and examples), but we still have to implement a way of determining whether a formula is true. As an example $\neg(x =_i x)$ is a well-stated formula, but it certainly thwarts our intuition. Looking at the mathematical practice we see that statements are accepted, if they can be proved, using previously accepted statements. As this all has to start somewhere, mathematicians tend to agree on a certain list of statements (called *axioms*) that are taken to be granted. In this section we will try to simulate this process on a formal basis by explicitly describing all the conclusions we consider to be logically evident. To simulate this we will introduce a relation $\vdash$ between sets $L \subseteq \text{form}(\mathcal{L})$ of formulae on the left-hand side (this shall be the set of axioms agreed upon) and formulae $\varphi \in \text{form}(\mathcal{L})$ on the right-hand side. If the relation $L \vdash \varphi$ is true, then φ can be proved on the basis of L. As always in formal logic we define this relation recursively, by presenting general truisms we consider evident and rules how to proceed from one formula to the next.
It has been decades of work of various mathematicians to establish formalisms, that are both easy to handle and sufficient for the purpose of mathematics. The following approach is due to U. Felgner who built up on the work of Frege. The reader who is interested in tracing the origins is asked to refer to Felgner's scripts on mathematical logic and set theory at the university of Tübingen.

Definition 14:
Let L and $M \subseteq \text{form}(\mathcal{L})$ be any sets of formulae of the many-sorted language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$, then we recursively define the relation

$$\vdash \quad \subseteq \quad \mathcal{P}(\text{form}(\mathcal{L})) \times \text{form}(\mathcal{L})$$

by letting $L \vdash \varphi$ iff φ can be derived from L by using the following axioms (L1),...,(L7) and deduction rules (D1),...,(D5) finitely many times. Thereby we speak of $L \vdash \varphi$ as φ *can be deduced from* L

(i) **Logical axioms:** For any term $t \in \text{term}(\mathcal{L})$, formulae φ, $\psi \in \text{form}(\mathcal{L})$ and any variable symbol $x \in \text{var}(\mathcal{L})$ of $\text{sort}(x) = i$ the following seven formulae of $\mathcal{L}$ are logical axioms:

(L1) $\emptyset \vdash \varphi \to (\psi \to \varphi)$

(L2) $\emptyset \vdash (\neg\varphi) \to (\varphi \to \psi)$

(L3) $\emptyset \vdash x =_i x$

(L4) $\emptyset \vdash (\forall_i\, x\, (\varphi \to \psi)) \to (\varphi \to \forall_i\, x\, \psi)$

(L5) $\emptyset \vdash \exists_i\, x_i\, \varphi \leftrightarrow \neg(\forall_i\, x\, (\neg\varphi))$

(L6) If x is freely substitutable by t in φ, then we admit the deduction

$$\emptyset \vdash (\forall_i\, x\, \varphi) \to \varphi[x : t]$$

(L7) Let $x, y \in \mathrm{var}(\mathcal{L})$ be two variables of the same sort i, formally $i := \mathrm{sort}(x) = \mathrm{sort}(y)$, such that x is freely substitutable by y in φ. And let φ' be derived from φ by substituting x by y at some free occurences only, then we allow

$$\emptyset \vdash (x =_i y) \to (\varphi \to \varphi')$$

Nota by φ' being derived from φ by substituting x by y at some free occurences we mean that $\forall k \in 1 \dots \ell(\varphi)$ we have

$$\varphi'[k] \quad = \quad \begin{cases} x \text{ or } y & \text{if } \varphi[k] = x \text{ is a free occurence of } x \\ \varphi[k] & \text{else} \end{cases}$$

(ii) **Deduction rules:** We now give five schemes how we can proceed from already known statements to new ones by means of logical deduction

(D1) **trivial rule:** For any $\lambda \in L$ we allow $L \vdash \lambda$

(D2) **dilution rule:** If $L \vdash \varphi$ then $L \cup M \vdash \varphi$

(D3) **modus ponens:** If $L \vdash \varphi$ and $L \vdash \varphi \to \psi$ then $L \vdash \psi$

(D4) **section rule:** If $L \cup \{\varphi\} \vdash \psi$ and $L \cup \{\neg\varphi\} \vdash \psi$ then $L \vdash \psi$

(D5) **generalization rule:** Let $x \in \mathrm{var}(\mathcal{L})$ be a variable of the sort $i := \mathrm{sort}(x)$ such that x is not both: a free variable of φ and of one of the $\lambda \in L$ - that is $x \notin \mathrm{free}(\varphi) \cap \bigcup\{\mathrm{free}(\lambda) \mid \lambda \in L\}$. If now $L \vdash \varphi$ then we also allow $L \vdash \forall_i \, x \, \varphi$.

(iii) **Tautologies:** For any set L of formulae we define the **consequences** of L to be the set of all formulae that can be deduced from L. And φ is called a **tautology**, iff it can be derived by applying the logical axioms (L1) to (L7) alone, i.e. we define the sets

$$\begin{aligned} \mathrm{con}(L) &:= \{\, \varphi \in \mathrm{form}(\mathcal{L}) \mid L \vdash \varphi \,\} \\ \mathrm{taut}(\mathcal{L}) &:= \mathrm{con}(\emptyset) \end{aligned}$$

(iv) **Logical equivalence:** Two formulae φ and ψ are called **logically equivalent**, iff the equivalence of φ and ψ can be deduced by applying (L1) to (L7) only. Equivalently iff ψ can be deduced from $\{\varphi\}$ and vice versa:

$$\begin{aligned} \varphi \dashv\vdash \psi \quad &:\Longleftrightarrow \quad \varphi \leftrightarrow \psi \in \mathrm{taut}(\mathcal{L}) \\ &\Longleftrightarrow \quad \{\varphi\} \vdash \psi \text{ and } \{\psi\} \vdash \varphi \end{aligned}$$

Now two sets of formulae $L, L' \subseteq \mathrm{form}(\mathcal{L})$ are called **logically equivalent modulo** $M \subseteq \mathrm{form}(\mathcal{L})$ combined with M they yield the same consequences, formally

$$L \dashv\vdash L' \ (\mathrm{mod}\ M) \quad :\Longleftrightarrow \quad \mathrm{con}(L \cup M) \;=\; \mathrm{con}(L' \cup M)$$

Remark 15:
The deduction rules are tailored is such a way that $\neg$ has the meaning of *not* and $\to$ has the meaning of *implies*. This really is the reasoning behind (D3). Then (L1) carries the idea, that if φ is true, then $\psi \to \varphi$ should be true, as well, regardless of the logical value of ψ. Likewise (L2) implements, that if φ is false, then $\varphi \to \psi$ should be true, as well, regardless of the logical value of ψ. In latin this ingenious idea has been given the name *ex falso quodlibet* - which translates into "out of falsehood everything can be concluded". The reasoning for this is that if φ never is true, then you will never need to check $\varphi \to \psi$ and hence never err when relying on $\varphi \to \psi$.

Lemma 16: Import-Export-Theorem
Consider any formal language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ and a set $L \subseteq \text{form}(\mathcal{L})$ and two single formulae φ, $\psi \in \text{form}(\mathcal{L})$ therein. The we have $L \vdash \varphi \to \psi$ if and only if we also have $L \cup \{\varphi\} \vdash \psi$. We can put this formally by using the equivalence sign of the meta-language:

$$L \vdash \varphi \to \psi \quad \Longleftrightarrow \quad L \cup \{\varphi\} \vdash \psi$$

Proof:
Let us first deal with the $\Longrightarrow$ part: By (D1) we have $L \cup \{\varphi\} \vdash \varphi$. And by the assumption and (D2) we also have $L \cup \{\varphi\} \vdash \varphi \to \psi$. But then (D3) allows to proceed to $L \cup \{\varphi\} \vdash \psi$. Now conversely for the $\Longleftarrow$ part: By (L2) we have $\emptyset \vdash (\neg\varphi) \to (\varphi \to \psi)$. As we have just seen, this implies $\{\neg\varphi\} \vdash \varphi \to \psi$. And by (D2) this also is

$$L \cup \{\neg\varphi\} \ \vdash \ \varphi \to \psi$$

Note that (L1) also yields $\emptyset \vdash \psi \to (\varphi \to \psi)$ and (D1) turns this into $L \cup \{\psi\} \vdash \psi \to (\varphi \to \psi)$. But by assumption we also have $L \cup \{\varphi\} \vdash \psi$ so modus ponens (D3) lets us proceed once again, to

$$L \cup \{\varphi\} \ \vdash \ \varphi \to \psi$$

But now the section rule (D4) springs into action, eliminating the appendix of L on the right-hand side altogether, leaving only $L \vdash \varphi \to \psi$. $\square$

The deduction rules here have been reduced to an absolute minimum in order to make as few assumptions as possible and to faciliate proofs by induction on the number of deduction steps used. The drawback is, that there are a lot of statements that are well-known or intuitively clear need to be proved. We will do so with some tautologies, to get a feel for the deduction rules. We could go on with this for dozens of pages but in the end we will forge a much more powerful tool to conduct proofs: Models. By the correctness and completeness theorems these two ways of conducting proofs will be equivalent and it is much easier to provide proofs in the setting of models.

Proposition 17:
Let again $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be any formal language and pick up any sets L, $M \subseteq \text{form}(\mathcal{L})$, a term $t \in \text{term}(\mathcal{L})$ of $\text{sort}(t) = i$, and φ, ψ, $\chi \in \text{form}(\mathcal{L})$ formulae therein. Then we obtain the following tautologies

(i) $\emptyset \vdash t =_i t$

(ii) $\emptyset \vdash \varphi \to \varphi$

(iii) $\emptyset \vdash \varphi \to \neg\neg\varphi$

(iv) $\emptyset \vdash \neg\neg\varphi \to \varphi$

(v) $\emptyset \vdash \varphi \vee (\neg\varphi)$

(vi) $\emptyset \vdash (\varphi \wedge \psi) \to (\psi \wedge \varphi)$

(vii) $\emptyset \vdash (\varphi \wedge \psi) \to \varphi$ and $\emptyset \vdash (\varphi \wedge \psi) \to \psi$

(viii) $\emptyset \vdash (\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$

(ix) $\emptyset \vdash (\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$

(x) $\emptyset \vdash (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi))$

(xi) If $L \vdash \varphi \to \psi$ and $M \vdash \psi \to \chi$ then $L \cup M \vdash \varphi \to \chi$

(xii) We have the deduction $L \vdash \varphi \wedge \psi$ if and only if we have both, the deductions $L \vdash \varphi$ and $L \vdash \psi$

Proof:

(i) Choose any variable $x \in \text{var}(\mathcal{L})$ of $\text{sort}(x) = i$ such that $x \notin \text{free}(t)$, then $\emptyset \vdash x =_i x$ by (L3) and (D5) turns this into $\emptyset \vdash \forall_i x\,(x =_i x)$. Also (L6) applied to $\varphi = (x =_i x)$ yields $\emptyset \vdash \forall_i x\,(x =_i x) \to (t =_i t)$ and from this we get $\emptyset \vdash t =_i t$ by an application of (D3).

(ii) By the trivial rule (D1) we have $\{\varphi\} \vdash \varphi$. Now the import-export-theorem 16 turns this deduction into $\emptyset \vdash \varphi \to \varphi$ already.

(v) Recall that by definition $\varphi \vee (\neg\varphi)$ is just another notation of the formula $(\neg\varphi) \to (\neg\varphi)$ and hence this has already been covered in (ii).

(iii) The axiom (L2) applied to $\psi := \neg\neg\varphi$ reads as $\emptyset \vdash \neg\varphi \to (\varphi \to \neg\neg\varphi)$ and therefore $\{\neg\varphi\} \vdash \varphi \to \neg\neg\varphi$. Likewise (L1) applied to $\varphi := \neg\neg\varphi$ and $\psi := \varphi$ reads as $\emptyset \vdash \neg\neg\varphi \to (\varphi \to \neg\neg\varphi)$ and hence $\{\neg\neg\varphi\} \vdash \varphi \to \neg\neg\varphi$. Now (D4) removes $\neg\varphi$ from the conditions and we are left with the second claim $\emptyset \vdash \varphi \to \neg\neg\varphi$.

(iv) By (L2) again we have $\emptyset \vdash (\neg\neg\varphi) \to ((\neg\varphi) \to \varphi)$. Using the import-export-theorem once we arrive at $\{\neg\neg\varphi\} \vdash \to (\neg\varphi) \to \varphi$. Using it again, we get $\{\neg\neg\varphi, \neg\varphi\} \vdash \varphi$. But there is no order in the set $\{\neg\neg\varphi, \neg\varphi\}$ of assumptions, so we can pull out $\neg\neg\varphi$ to arrive at

$$\{\neg\varphi\} \vdash (\neg\neg\varphi) \to \varphi$$

Next we use (L1) to create $\emptyset \vdash \varphi \to ((\neg\neg\varphi) \to \varphi)$ and the import-export-theorem to turn this into $\{\varphi\} \vdash (\neg\neg\varphi) \to \varphi$. Combining the last two formula in the section rule we conclude $\emptyset \vdash (\neg\neg\varphi) \to \varphi$.

(xi) Using the import-export theorem we can turn $L \vdash \varphi \to \psi$ into $L \cup \{\varphi\} \vdash \psi$. And by the dilution rule this also is $L \cup M \cup \{\varphi\} \vdash \psi$. But this rule also turns $M \vdash \psi \to \chi$ into $L \cup M \cup \{\varphi\} \vdash \psi \to \chi$. So modus ponens yields $L \cup M \cup \{\varphi\} \vdash \chi$. Applying import-export the other way around, we see $L \cup M \vdash \varphi \to \chi$.

(viii) The idea of the proof is to start with the set $\{\varphi \to \psi, \neg\psi, \varphi\}$ of assumptions, to eliminate φ and deduce $\neg\psi \to \neg\varphi$ via import-export. So let us take one step at a time:

(A)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash \varphi$	by (D1)
(B)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash \varphi \to \psi$	by (D1)
(C)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash \psi$	by (D3) + (A) + (B)
(D)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash \neg\psi$	by (D1)
(E)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash (\neg\psi) \to (\psi \to \neg\varphi)$	by (L2) + (D2)
(F)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash \psi \to \neg\varphi$	by (D3) + (D) + (E)
(G)	$\{\varphi \to \psi, \neg\psi, \varphi\} \vdash \neg\varphi$	by (D3) + (C) + (F)
(H)	$\{\varphi \to \psi, \neg\psi, \neg\varphi\} \vdash \neg\varphi$	by (D1)
(I)	$\{\varphi \to \psi, \neg\psi\} \vdash \neg\varphi$	by (D4) + (G) + (H)
(J)	$\{\varphi \to \psi\} \vdash \neg\psi \to \neg\varphi$	by lemma 16
(K)	$\emptyset \vdash (\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$	by lemma 16

(ix) The idea of the proof is similar to the one before: We start with the set $\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\}$ of assumptions, to eliminate $\neg\psi$ and deduce $\varphi \to \psi$ via import-export. The double negation $\neg\neg\varphi$ is covered by what we have proved (iv) already. So let us take one step at a time:

(A)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \neg\psi$	by (D1)
(B)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \neg\psi \to \neg\varphi$	by (D1)
(C)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \neg\varphi$	by (D3) + (A) + (B)
(D)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \varphi$	by (D1)
(E)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \varphi \to \neg\neg\varphi$	by (iv) + (D2)
(F)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \neg\neg\varphi$	by (D3) + (D) + (E)
(G)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash (\neg\neg\varphi) \to (\neg\varphi \to \psi)$	by (L2) + (D2)
(H)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \neg\varphi \to \psi$	by (D3) + (F) + (G)
(I)	$\{\neg\psi \to \neg\varphi, \varphi, \neg\psi\} \vdash \psi$	by (D3) + (C) + (H)
(J)	$\{\neg\psi \to \neg\varphi, \varphi, \psi\} \vdash \psi$	by (D1)
(K)	$\{\neg\psi \to \neg\varphi, \varphi\} \vdash \psi$	by (D4) + (G) + (H)
(M)	$\{\neg\psi \to \neg\varphi\} \vdash \varphi \to \psi$	by lemma 16
(N)	$\emptyset \vdash (\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$	by lemma 16

(vi) Recall that by definition $\varphi \wedge \psi$ is nothing but $\neg(\varphi \to (\neg\psi))$. So let us define the formula $\omega := (\psi \to (\neg\varphi))$, then $\neg\omega$ is nothing but $\psi \wedge \varphi$. By (L1) we have $\{\,\omega\,\} \vdash \omega = \psi \to (\neg\varphi)$. And by (viii) and (D2) we also have $\{\,\omega\,\} \vdash (\psi \to \neg\varphi) \to (\neg\neg\varphi \to \neg\psi)$. Using modus ponens we conclude $\{\,\omega\,\} \vdash \neg\neg\varphi \to \neg\psi$. By (iii) and (D2) we also have $\{\,\omega\,\} \vdash \varphi \to \neg\neg\varphi$. So applying (x) we conclude $\{\,\omega\,\} \vdash \varphi \to \neg\psi$. Import-export once again turns this into $\emptyset \vdash \omega \to (\varphi \to \neg\psi)$. This relation combined with another dose of (viii) and modus ponens yields $\emptyset \vdash \neg(\varphi \to \neg\psi) \to \neg\omega$. But the formula on the left side of $\to$ is $\varphi \wedge \psi$ and by construction $\neg\omega$ is $\psi \wedge \varphi$.

(vi) Recall that by definition $\varphi \wedge \psi$ is nothing but $\neg(\varphi \to (\neg\psi))$. So let us use (L1) to create the formula

$$\emptyset \vdash (\neg\varphi) \to \big(\psi \to (\neg\varphi)\big)$$

If we set $\alpha := \neg\varphi$ and $\beta := (\psi \to (\neg\varphi))$ this is nothing but $\emptyset \vdash \alpha \to \beta$. And by (vii) we have $\emptyset \vdash (\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$. Using modus ponens this yields $\emptyset \vdash \neg\beta \to \neg\alpha$. This is just

$$\emptyset \vdash \neg\big(\psi \to (\neg\varphi)\big) \to (\neg\neg\varphi)$$

Note that this is $\emptyset \vdash (\psi \wedge \varphi) \to (\neg\neg\varphi)$ already. Using import-export this becomes $\{\,\psi \wedge \varphi\,\} \vdash \neg\neg\varphi$. But we also have (iv) + dilution, which tells us, that $\{\,\psi \wedge \varphi\,\} \vdash \neg\neg\varphi \to \varphi$ so together with modus ponens we are allowed to proceed to $\{\,\psi \wedge \varphi\,\} \vdash \varphi$. But with import-export again

$$\emptyset \vdash (\psi \wedge \varphi) \to \varphi$$

Interchanging the names of φ and ψ we already have $\emptyset \vdash (\varphi \wedge \psi) \to \psi$. And by (vii) we also have $\emptyset \to (\varphi \wedge \psi) \to (\psi \wedge \varphi)$. So if we apply (xi) to the formula of (vii) and the formula above we find $\emptyset \vdash (\varphi \wedge \psi) \to \varphi$.

(x) Let us abbreviate $\omega := \varphi \to (\psi \to \chi)$ and $L := \{\,\omega, \varphi \to \psi, \varphi\,\}$. Using this set of formulae we perform one more formal deduction: This time we get

(A)	$L \vdash \varphi$	by (D1)
(B)	$L \vdash \varphi \to \psi$	by (D1)
(C)	$L \vdash \psi$	by (D3) + (A) + (B)
(D)	$L \vdash \omega = \varphi \to (\psi \to \chi)$	by (D1)
(E)	$L \vdash \psi \to \chi$	by (D3) + (A) + (D)
(F)	$L \vdash \chi$	by (D3) + (C) + (E)
(G)	$\{\,\omega, \varphi \to \psi\,\} \vdash \varphi \to \chi$	by lemma 16
(H)	$\{\,\omega\,\} \vdash (\varphi \to \psi) \to (\varphi \to \chi)$	by lemma 16
(I)	$\emptyset \vdash \omega \to (\varphi \to \psi) \to (\varphi \to \chi)$	by lemma 16

(xii) Suppose we have $L \vdash \varphi \wedge \psi$, as by (vii) and (D2) we always have $L \vdash (\varphi \wedge \psi) \to \varphi$ modus ponens immediately yields $L \vdash \varphi$. Likewise we then get $L \vdash \psi$ from (vii). Conversely, let us assume we have both $L \vdash \varphi$ and $L \vdash \psi$. Then we need to show $L \vdash \varphi \wedge \psi$. If we let $\omega := (\varphi \to \neg\psi)$ this is $L \vdash \neg\omega$.

By the trivial rule (D1) we have $L \cup \{\omega\} \vdash \omega = \varphi \to \neg\psi$. And as we have $L \vdash \varphi$ by assumption (D2) tells us $L \cup \{\omega\} \vdash \varphi$. Together with modus ponens this is $L \cup \{\omega\} \vdash \neg\psi$. Import-export turns this into $L \vdash \omega \to \neg\psi$. Once again (vii) and (D2) give rise to the formula $L \vdash (\omega \to \neg\psi) \to (\neg\neg\psi \to \neg\omega)$ and hence modus ponens allows us to proceed to $L \vdash \neg\neg\psi \to \neg\omega$. But we also have $\emptyset \vdash \psi \to \neg\neg\psi$ according to (iii) and hence (xi) turns these formulae into $L \vdash \psi \to \neg\omega$. Now our assumption $L \vdash \psi$ comes in handy, as modus ponens helps us to conclude $L \vdash \neg\omega$.

□

Proposition 18:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a many-sorted formal language. Consider any three formulae φ, ψ and $\chi \in \text{form}(\mathcal{L})$ of this language. Then the following four statements are equivalent

(a) $\emptyset \vdash \varphi \to (\psi \to \chi)$

(b) $\emptyset \vdash \psi \to (\varphi \to \chi)$

(c) $\emptyset \vdash (\varphi \wedge \psi) \to \chi$

(d) $\{\varphi, \psi\} \vdash \chi$

Nota however that the junctor $\to$ is *not* associative, that is the deduction $\emptyset \vdash (\varphi \to \psi) \to \chi$ is *not* equivalent to (a). In fact $\emptyset \vdash (\varphi \to \psi) \to \chi$ is stronger in the sense that it implies (a), but not vice verse.

Proof:
For (a) $\iff$ (d) we simply use the import-export theorem twice: This theorem tells us that (a) is equivalent to $\{\varphi\} \vdash \psi \to \chi$. Another application and this is equivalent to (d).

For (b) $\iff$ (d) we just have to note, that $\{\varphi, \psi\} = \{\psi, \varphi\}$ so starting with (d) we can pull out φ first, to find $\{\psi\} \vdash \varphi \to \chi$ and ψ in a second application, to find (b).

So let us prove (a) $\implies$ (c) next, however we will prove (c) in the form $\{\varphi \wedge \psi\} \vdash \chi$ which is equivalent to (c), due to the import-export-theorem

$$\begin{array}{lll}
(A) & \{\varphi \wedge \psi\} \vdash \varphi \wedge \psi & \text{by (D1)} \\
(B) & \{\varphi \wedge \psi\} \vdash \varphi \wedge \psi \to \varphi & \text{by 17.(vii) + (D2)} \\
(C) & \{\varphi \wedge \psi\} \vdash \varphi & \text{by (D3) + (A) + (B)} \\
(D) & \{\varphi \wedge \psi\} \vdash \varphi \to (\psi \to \chi) & \text{by (a) + (D2)} \\
(E) & \{\varphi \wedge \psi\} \vdash \psi \to \chi & \text{by (D3) + (C) + (D)} \\
(F) & \{\varphi \wedge \psi\} \vdash \varphi \wedge \psi \to \psi & \text{by 17.(vii) + (D2)} \\
(G) & \{\varphi \wedge \psi\} \vdash \chi & \text{by (D3) + (A) + (F) + (E)}
\end{array}$$

To come full circle, we now prove (c) $\implies$ (d): By the trivial rule we have $\{\varphi, \psi\} \vdash \varphi$ and $\{\varphi, \psi\} \vdash \psi$. So by 17.(x) this is $\{\varphi, \psi\} \vdash \varphi \wedge \psi$. Together with the assumption (c) modus ponens turns this into $\{\varphi, \psi\} \vdash \chi$, which is (d). □

The following properties of the equality relations $=_i$ only are what is to be expected. If we could not prove them, we would have to include them in our list of axioms. They will be needed for Henkin's Lemma 45 which in turn is required for the Completeness Theorem 39.(iv). It turns out, that these properties already follow automatically however, so here's why: [For the proof we will use results of section 5 already, but thematically these properties belong here].

Proposition 19:
If $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$ is a many-sorted formal language and r, s and $t \in \mathrm{term}(\mathcal{L})$ are terms of the $\mathrm{sort}(r) = \mathrm{sort}(s) = \mathrm{sort}(t) = i \in I$, then we find the following tautologies

(i) $\emptyset \vdash t =_i t$

(ii) $\emptyset \vdash (s =_i t) \to (t =_i s)$

(iii) $\emptyset \vdash (r =_i s) \to (s =_i t \to r =_i t)$

(iv) $\emptyset \vdash \exists_i x\,(x =_i x)$ for any $x \in \mathrm{var}(\mathcal{L})$ of $\mathrm{sort}(x) = i$

(v) If $x \in \mathrm{var}(\mathcal{L})$ is a variable symbol of $\mathrm{sort}(x) = \mathrm{sort}(t)$ and $x \notin \mathrm{free}(t)$ then $\emptyset \vdash \exists_i x\,(x =_i t)$

(vi) If $\varphi \in \mathrm{form}(\mathcal{L})$ is a formulae and $x \in \mathrm{var}(\mathcal{L})$ is freely substitutable by t in φ, then we have

$$\emptyset \vdash \varphi[x : t] \to \exists_i x\, \varphi$$

(vii) If x, $y \in \mathrm{var}(\mathcal{L})$ are variables of the same $\mathrm{sort}(x) = \mathrm{sort}(y) = j \in I$ such that $x \in \mathrm{free}(t)$ and $t' \in \mathrm{term}(\mathcal{L})$ is a term that is derived from t by replacing x by y at some (or all) instances, then

$$\emptyset \vdash (x =_j y) \to (t =_i t')$$

Now consider the terms $s_1, \ldots, s_n$ and $t_1, \ldots, t_n \in \mathrm{term}(\mathcal{L})$ where $1 \le n \in \mathbb{N}$ of the sorts $i_k = \mathrm{sort}(s_k) = \mathrm{sort}(t_k)$. Then we introduce a notation for the n-fold conjunction by

$$\bigwedge_{k=1}^{n} (s_k =_{i_k} t_k) \;:=\; (s_1 =_{i_1} t_1) \wedge (s_2 =_{i_2} t_2) \wedge \cdots \wedge (s_n =_{i_n} t_n)$$

If now f is a function symbol of $\mathcal{L}$ of $\mathrm{sort}(f) = (i_1, i_2, \ldots, i_n, i_{n+1})$ appropriate to the terms above, then we obtain the following tautology (viii), that expresses the right-uniqueness of functions

$$\emptyset \vdash \bigwedge_{k=1}^{n} (s_k =_{i_k} t_k) \to f(s_1, s_2, \ldots, s_n) =_{i_{n+1}} f(t_1, t_2, \ldots, t_n)$$

And if $\varphi \in \mathrm{form}(\mathcal{L})$ is a formula, $x_1, \ldots, x_n \in \mathrm{var}(\mathcal{L})$ are variable symbols of $\mathrm{sort}(x_k) = i_k$ such that x_k is freely substitutable by both s_k and by t_k in φ, then we get another tautology (ix)

$$\emptyset \vdash \bigwedge_{k=1}^{n} (s_k =_{i_k} t_k) \to \big(\varphi[x_1 : s_1, \ldots, x_n : s_n] \to \varphi[x_1 : t_1, \ldots, x_n : t_n]\big)$$

Intuitively speaking (iv) expresses that in any sort there is something to talk about, there is no variable that does not exist. Likewise (v) tells us that any term has a result. Conversely by (vi): If φ is satisfied by a term t then there is some value that satisfies φ. All these tautologies guarantee that the language's content is not void.

Proof:

(i) Has already been proved in 17.(i).

(ii) Let x, $y \in \mathrm{var}(\mathcal{L})$ be any two variable symbols of $\mathrm{sort}(x) = \mathrm{sort}(y) = i$, that do *not* appear in $\mathrm{free}(s =_i t)$. Now take $\varphi = (x =_i x)$ and $\varphi' = (y =_i x)$ in (L7), then we begin with

$$\begin{array}{rll}
(A) & \emptyset \vdash x =_i y \to (x =_i x \to y =_i x) & \text{by (L7)} \\
(B) & \{x =_i y, x =_i x\} \vdash y =_i x & \text{by lemma 16} \\
(C) & \emptyset \vdash x =_i x \to (x =_i y \to y =_i x) & \text{by lemma 16} \\
(D) & \emptyset \vdash x =_i x & \text{by (L3)} \\
(E) & \emptyset \vdash x =_i y \to y =_i x & \text{by (D3) + (C) + (D)} \\
(F) & \emptyset \vdash \forall_i x\, (x =_i y \to y =_i x) & \text{by (D5) + (E)}
\end{array}$$

Let us abbreviate the formula $\psi := (x =_i y \to y =_i x)$ then $\psi[x : s]$ is $s =_i y \to y =_i s$ and we may employ (L6) to find

$$\begin{array}{rll}
(G) & \emptyset \vdash \forall_i x\, \psi \to \psi[x : s] & \text{by (L6)} \\
(H) & \emptyset \vdash \psi[x : s] \;=\; (s =_i y \to y =_i s) & \text{by (D3) + (F) + (G)} \\
(I) & \emptyset \vdash \forall_i y\, (s =_i y \to y =_i s) & \text{by (D5) + (H)}
\end{array}$$

Analogous to the above let $\chi = (s =_i y \to y =_i s)$ then clearly $\chi[y : t]$ is $s =_i t \to t =_i s$ so with (L6) again we conjure up

$$\begin{array}{rll}
(J) & \emptyset \vdash \forall_i y\, \chi \to \chi[y : t] & \text{by (L6)} \\
(K) & \emptyset \vdash \chi[y : t] \;=\; (s =_i t \to t =_i s) & \text{by (D3) + (I) + (J)}
\end{array}$$

(iii) Let x, y and $z \in \mathrm{var}(\mathcal{L})$ be any three variable symbols, all of sort i, that do *not* appear in $\mathrm{free}(r) \cap \mathrm{free}(s) \cap \mathrm{free}(t)$. Now take $\varphi = (y =_i z)$ and $\varphi' = (x =_i z)$ in (L7), then we begin with

$$\begin{array}{rll}
(A) & \emptyset \vdash y =_i x \to (y =_i z \to x =_i z) & \text{by (L7)} \\
(B) & \emptyset \vdash x =_i y \to y =_i x & \text{by (ii)} \\
(C) & \emptyset \vdash x =_i y \to (y =_i z \to x =_i z) & \text{by 17.(xi) + (B) + (A)} \\
(D) & \emptyset \vdash \forall_i x\, \big(x =_i y \to (y =_i z \to x =_i z)\big) & \text{by (D5) + (C)} \\
(E) & \emptyset \vdash r =_i y \to (y =_i z \to r =_i z) & \text{by (L6) + (D3)} \\
(F) & \emptyset \vdash \forall_i y\, \big(r =_i y \to (y =_i z \to r =_i z)\big) & \text{by (D5) + (E)} \\
(G) & \emptyset \vdash r =_i s \to (s =_i z \to r =_i z) & \text{by (L6) + (D3)} \\
(H) & \emptyset \vdash \forall_i z\, \big(r =_i s \to (s =_i z \to r =_i z)\big) & \text{by (D5) + (G)} \\
(I) & \emptyset \vdash r =_i s \to (s =_i t \to r =_i t) & \text{by (L6) + (D3)}
\end{array}$$

(iv) Let us abbreviate $x := x_{i,j}$ and $\varphi := \neg(x =_i x)$. Then we have $\emptyset \vdash \forall_i\, x\, \varphi \to \varphi[x : x]$ by (L6). Now as $\varphi[x : x] = \varphi$ 24.(viii) turns this into $\emptyset \vdash \neg\varphi \to \neg\forall_i\, x\, \varphi$, which is

$$\emptyset \vdash \neg\neg(x =_i x) \to \neg\forall_i\, x\, \neg(x =_i x)$$

Clearly $\emptyset \vdash (x =_i x) \to \neg\neg(x =_i x)$ by 17.(iii) and hence $\emptyset \vdash (x =_i x) \to \neg\forall_i\, x\, \neg(x =_i x)$ by 17.(xi). Yet $\emptyset \to (x =_i x)$ is the logical axiom (L3) so modus ponend (D3) takes us to $\emptyset \vdash \neg\forall_i\, x\, \neg(x =_i x)$ which is the claim $\emptyset \vdash \exists_i\, x\, (x =_i x)$ by axiom (L5).

(vi) By (L6) we get $\emptyset \vdash \forall_i\, x\, \psi \to \psi[x : t]$. If we take $\psi := \neg\varphi$ this reads as $\emptyset \vdash (\forall_i\, x\, \neg\varphi) \to \neg\varphi[x : t]$. And by 24.(viii) this can be turned into

$$\emptyset \vdash \neg\neg\varphi[x : t] \to \neg(\forall_i\, x\, (\neg\varphi))$$

But $\neg\neg\varphi[x : t] \dashv\vdash \varphi[x : t]$ by 24.(iv) and $\neg\forall_i\, x\, \neg\varphi \dashv\vdash \exists_i\, x\, \varphi$ by rule (L5) so we can invoke 23.(ii) to find the claim $\emptyset \vdash \varphi[x : t] \to \exists_i\, x\, \varphi$.

(v) Consider the formula $\varphi = (x =_i t)$, then x is freely substitutable by t in φ and $\pi[x : t] = (t =_i t)$, thus we have $\emptyset \vdash \varphi[x : t]$ by (i). But we also have $\emptyset \vdash \varphi[x : t] \to \exists_i\, x\, \varphi$ by (vi) so modus ponens (D3) allows the conclusion $\emptyset \vdash \exists_i\, x\, \varphi$ which is the claim.

(vii) Let $\varphi := (t =_i t)$ and $\varphi' := (t = t')$. Then φ' is derived from φ by replacing x by y at some free occurences. Thus by (L6) we have

$$\emptyset \vdash (x =_j y) \to (\varphi \to \varphi')$$

Using import export twice we find $\{ x =_j y, \varphi \} \vdash \varphi'$ and pulling these formula out again in reversed order we get $\emptyset \vdash \varphi \to ((x =_j y) \to \psi'$. But we also have $\emptyset \vdash t =_i t$ (which is $\emptyset \vdash \varphi$) by (i) so modus ponens takes us to the claim - recall $\varphi' = (t =_i t')$

$$\emptyset \vdash (x =_j y) \to \varphi'$$

(viii) Pick up pairwise distinct variables $x_1, \dots, x_n$ and $y_1, \dots, y_n$ such that $\mathrm{sort}(x_k) = i_k = \mathrm{sort}(y_k)$ and none of these variables appears in any of the terms $s_1, \dots, s_n, t_1, \dots, t_n$. Then we get by (vii)

$$\emptyset \vdash \varphi_0 := (x_1 =_{i_1} y_1) \to \big(f(x_1, x_2, \dots, x_n) =_{i_{n+1}} f(y_1, x_2, \dots, x_n)\big)$$

Now we use the generalization rule (L5) to build up $\emptyset \vdash \forall_{i_1}\, x_1\, \varphi_0$ from this formula and use (L6) to create $\emptyset \vdash \forall_{i_1}\, x_1\, \varphi_0 \to \varphi_0[x_1 : s_1]$ and apply modus ponens afterwards to arrive at

$$\emptyset \vdash \psi_1 := (s_1 =_{i_1} y_1) \to \big(f(s_1, x_2, \dots, x_n) =_{i_{n+1}} f(y_1, x_2, \dots, x_n)\big)$$

The same sequel of arguments - build up $\emptyset \vdash \forall_{i_1}\, y_1\, \psi_1$ with (L5), invoke (L6) $\emptyset \vdash \forall_{i_1}\, y_1\, \psi_1 \to \psi_1[y_1 : t_1]$ and modus ponens - yields

$$\emptyset \vdash \varphi_1 := (s_1 =_{i_1} t_1) \to \big(f(s_1, x_2, \dots, x_n) =_{i_{n+1}} f(t_1, x_2, \dots, x_n)\big)$$

Let $\varphi_1' := (s_1 =_{i_1} t_1) \to (f(s_1, x_2, \dots, x_n) =_{i_{n+1}} f(t_1, y_2, x_3, \dots, x_n))$, that is we replaced x_2 by y_2 in its second occurence. Then we have $\emptyset \vdash (x_2 =_{i_2} y_2) \to (\varphi_1 \to \varphi_1')$. But as in (vii) before we may import twice, swap and export twice the interchange the sequel of premisses. That is $\emptyset \vdash \varphi_1 \to ((x_2 =_{i_2} y_2) \to \varphi_1')$. And as we have φ_1 already we can use modus ponens to get

$$\emptyset \vdash \psi_2 := (x_2 =_{i_2} y_2) \to \varphi_1'$$

We now iterate the arguments that took us from φ_0 to φ_1 but this time we turn x_2 into s_2 and y_2 into t_2 in ψ_2. Doing this we arrive at

$$\emptyset \vdash (s_2 =_{i_2} t_2) \to \big((s_1 =_{i_1} t_1) \to \varepsilon_2\big)$$

where $\varepsilon_2 = (f(s_1, s_2, x_3, \dots, x_n) =_{i_{n+1}} f(t_1, t_2, x_3, \dots, x_n)$. Clearly we can go on like this indefinitely, always replacing x_k by s_k and y_k by t_k. In the end we will arrive at

$$\emptyset \vdash (s_n =_{i_n} t_n) \to \Big(\dots (s_2 =_{i_2} t_2) \to \big((s_1 =_{i_1} t_1) \to \varepsilon_n\big)\Big)$$

where $\varepsilon_n = f(s_1, \dots, s_n) =_{i_{n+1}} f(t_1, \dots, t_n)$. But clearly this formula can be reformulated as the one in the claim, using 18 and induction.

(ix) This proof follows precisely the same lines, as the proof of (viii). First of all pick up variable symbols $y_1, \dots, y_n \in \mathrm{var}(\mathcal{L})$ that do not appear in φ and any s_k and t_k at all. Also let $\varphi_1 := \varphi = \varphi(x_1, \dots, x_n)$ and $\varphi_1' := \varphi_1[x_1 : y_1] = \varphi(y_1, x_2, \dots, x_n)$ then (L7) allows us to commence with

$$\emptyset \vdash (x_1 =_{i_i} y_1) \to (\varphi(x_1, x_2, \dots, x_n) \to \varphi(y_1, x_2, \dots, x_n))$$

This formula takes the place of φ_0 in the proof of (viii). Again we can start our machinery to replace x_1 by s_1 and y_1 by t_1, then x_2 by y_2, in turn x_2 by s_2 and y_2 by t_2 and so on until we arrive at

$$\emptyset \vdash (s_n =_{i_n} t_n) \to \Big(\dots (s_2 =_{i_2} t_2) \to \big((s_1 =_{i_1} t_1) \to \iota_n\big)\Big)$$

where $\iota_n = \varphi[x_1 : s_1, \dots, x_n : s_n] \to \varphi[x_1 : t_1, \dots, x_n : t_n]$. Again this formula can be reformulated as the one in the claim, using 18 and induction.

□

Corollary 20:
As always let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$ be a formal language, but now let $1 \le n \in \mathbb{N}$ and say $\varphi_1, \dots, \varphi_n$ and $\psi \in \mathrm{form}(\mathcal{L})$ are formulae in the language $\mathcal{L}$. Then the following two statements are equivalent:

(a) $\emptyset \vdash (\varphi_1 \wedge \dots \wedge \varphi_n) \vdash \psi$

(b) $\{\varphi_1, \dots, \varphi_n\} \vdash \psi$

Nota from a formal point of view we have to introduce $\varphi_1 \wedge \dots \wedge \varphi_n$ by recursion $(\varphi_1 \wedge \dots \wedge \varphi_{n-1}) \wedge \varphi_n$ but the reader is most probably well aware of the fact, that the junctor $\wedge$ is associative (that is $(\varphi \wedge \psi) \wedge \chi$ and $\varphi \wedge (\psi \wedge \chi)$ are logically equivalent) such that any order of applying brackets to $\varphi_1 \wedge \dots \wedge \varphi_n$ would result in a logically equivalent formula. Hence it is customary to omit the bracketing.

Proof:
For this proof, we use induction on n: The case $n = 1$ is just the import-export-theorem 16 for $L = \emptyset$. The case $n = 2$ is just the equivalence of (c) and (d) in proposition 18. So we concern ourselves with $n \geq 3$ only: Let us abbreviate $\varphi := \varphi_1 \wedge \cdots \wedge \varphi_{n-1}$, then by 18 we know, that (a)

$$\emptyset \vdash (\varphi \wedge \varphi_n) \to \psi$$

is equivalent, to $\{ \varphi, \varphi_n \} \vdash \psi$. We now use the import-export-theorem twice: This again is equivalent to $\{ \varphi \} \vdash \varphi_n \to \psi$ which in turn is equivalent to $\emptyset \vdash \varphi \to (\varphi_n \to \psi)$. Let us insert the abbreviation of φ, then we have found another equivalence to (a)

$$\emptyset \vdash (\varphi_1 \wedge \cdots \wedge \varphi_{n-1}) \to (\varphi_n \to \psi)$$

Here the set of assumptions only contains $n-1$ elements and hence we may use the induction hypothesis to reformulate this formula equivalently into

$$\{ \varphi_1, \ldots, \varphi_{n-1} \} \vdash \varphi_n \to \psi$$

The rest is another dose of the import-export-theorem: We can pull φ_n into the assumptions, to arrive at the formula (b) in the claim: $\{ \varphi_1, \ldots, \varphi_{n-1}, \varphi_n \} \vdash \psi$. □

Corollary 21:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a formal language and $\vdash$ the relation of deduction of $\mathcal{L}$. That is $\vdash$ is a relation between subsets $L \subseteq \text{form}(\mathcal{L})$ and elements $\varphi \in \text{form}(\mathcal{L})$. Then for any sets L, $M \subseteq \text{form}(\mathcal{L})$ of formulae and any formulae λ, $\varphi \in \text{form}(\mathcal{L})$ we have the properties

(1) If $\lambda \in L$ then $L \vdash \lambda$

(2) If $L \vdash \varphi$ then there already is a finite subset $L_0 \subseteq L$ such that $L_0 \vdash \varphi$

(3) If $\forall \lambda \in L : M \vdash \lambda$ and $L \vdash \varphi$, then we also have $M \vdash \varphi$

Proof:
Property (1) is just the trivial rule (D1) of the deduction $\vdash$. Property (2) is clear: If we have $L \vdash \varphi$, then we used the axioms and rules of deduction finitely many times. The only rule that truly requires a formula $\lambda \in L$ is (D1) and this has been used finitely many times, only. Hence we may take L_0 to be the set of all $\lambda \in L$ that occurred in the deduction $L \vdash \varphi$ when using rule (D1). And this is finite. It remains to prove property (3):

As $L \vdash \varphi$ and (2) there are finitely many formulae $\lambda_1, \ldots, \lambda_n \in L$ such that $\{ \lambda_1, \ldots, \lambda_n \} \vdash \varphi$. By corollary 20 this can be reformulated, as $\emptyset \vdash (\lambda_1 \wedge \cdots \wedge \lambda_n) \vdash \varphi$. Now by the dilution rule (D2)

$$M \vdash (\lambda_1 \wedge \cdots \wedge \lambda_n) \vdash \varphi$$

On the other hand, we have $M \vdash \lambda_i$ for any $i \in 1 \dots n$, by assumption. Now by (obvious induction on) 17.(xii) we also find that M allows to deduce

$$M \vdash \lambda_1 \wedge \dots \wedge \lambda_n$$

Combining these two deductions under modus ponens (D3) we find $M \vdash \varphi$, which had to be shown. $\square$

Deduction is not quite a Dependence Relation

In other words, the above corollary states that $\vdash$ satisfies the first 3 properties of a *dependence relation*. The relation of linear dependence is the role-model for these sort of relations. For the general notion refer to [7] problems to section 19 or [9] section 3.6. There is one property left, that a dependence relation would have to satisfy. If this was true, then we could conclude that there are $\vdash$bases for any set $\mathrm{con}(L)$ and that all bases of $\mathrm{con}(L)$ share the same cardinality, and so on. Alas this is not the case. The missing property reads as: For any $L \subseteq \mathrm{form}(\mathcal{L})$ and any two formulae $\varphi, \psi \in \mathrm{form}(\mathcal{L})$ we would like to have

$$L \cup \{\varphi\} \vdash \psi \text{ and } L \not\vdash \psi \quad \Longrightarrow \quad L \cup \{\psi\} \vdash \varphi$$

This is untrue in general, however: To give an idea, let L be the set that contains the single statement *If x is a dog, then x barks*. And let φ be *x is a dog* and ψ be *x barks*. If we have L and φ then ψ is clear by modus ponens (any dog barks). However from L and ψ we cannot conclude φ. It may well be that there are other animals, that bark, not only dogs.

To give a formally correct counter-example we jump ahead: Using the completeness and correctness theorems 39.(iv), we know that deduction $L \vdash \varphi$ and implication $L \models \varphi$ are equivalent. So it suffices to disprove $L \models \varphi$ to have $L \not\vdash \varphi$. And this can be done by finding a single realization, that acts as a counter-example: Let $\mathcal{L}$ be the one-sorted language, that only contains two 1-ary relation symbols: D and B. We think of $D(x)$ as *x is a dog* and of $B(x)$ as *x barks*. Let also

$$\begin{aligned} L &:= \{\forall x (D(x) \to B(x))\} \\ \varphi &:= \forall x\, D(x) \\ \psi &:= \forall x\, B(x) \end{aligned}$$

Then $L \cup \{\varphi\} \vdash \psi$ by modus ponens, and $L \vdash \psi$ is untrue: Suppose we had $L \vdash \psi$, then $L \models \psi$. But we find a realization (X, ϱ) of $\mathcal{L}$, that satisfies L: Let $X := \mathbb{Z}$ and $\varrho(D) := 4\mathbb{Z}$ and $\varrho(B) := 2\mathbb{Z}$. That is $D(x)$ is $4 \mid x$ and $B(x)$ amounts to $2 \mid x$. And for any $x \in \mathbb{Z}$ the implication $4 \mid x \implies 2 \mid x$ is true. However ψ is untrue in (X, ϱ), e.g. for $x = 1$ we do not have $B(x)$. Hence $L \models \psi$ is untrue, such that $L \not\vdash \psi$, as well.

That is L, φ and ψ satisfy the conditions $L \cup \{\varphi\} \vdash \psi$ and $L \not\vdash \psi$. Let us prove, that the conclusion $L \cup \{\psi\} \vdash \varphi$ is false, however. Again we present a realisation (X, ϱ) of $\mathcal{L}$ that satisfies $L \cup \{\psi\}$ but not φ. This time $X := \mathbb{Z}$, $\varrho(D) := 2\mathbb{Z}$ and $\varrho(B) := \mathbb{Z}$. Then ψ is clear, as $\varrho(B) = X$ and L is clear, as $2\mathbb{Z} \subseteq \mathbb{Z}$. However φ is not satisfied, as $1 \notin \varrho(D)$, for example. Again we conclude $L \cup \{\psi\} \not\vdash \varphi$ from the falsification of $L \cup \{\psi\} \models \varphi$.

5 First Results

We have put quite some effort into defining (many-sorted, first order) languages and the notion of deduction in these. It is time to reap (part of) the harvest. In this section we will start with some easy equivalences. Afterwards we provide some neat consequences of the theory that are hard to come by (or rather that are self-evident but hard to argue for), yet commonplace in mathematics.

Corollary 22:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be any formal language and φ and $\psi \in \text{form}(\mathcal{L})$ be formulae therein. Then the following three statements are equivalently saying that φ and ψ are logically equivalent

(a) $\varphi \;\dashv\vdash\; \psi$

(b) $\neg\varphi \;\dashv\vdash\; \neg\psi$

(c) $\emptyset \vdash \varphi \to \psi$ and $\emptyset \vdash \psi \to \varphi$

(d) For any set of formulae $L \subseteq \text{form}(\mathcal{L})$ the deduction $L \vdash \varphi$ is possible if and only if the deduction $L \vdash \psi$ is possible, too. Formally

$$L \vdash \varphi \iff L \vdash \psi$$

Nota that the due to property (c) the relation $\dashv\vdash$ thereby becomes an equivalence relation on the set of formulae $\text{form}(\mathcal{L})$ of the language $\mathcal{L}$. That is we have (1) $\varphi \dashv\vdash \varphi$ for any formula φ, (2) $\varphi \dashv\vdash \psi$ is equivalent to $\psi \dashv\vdash \varphi$ and (3) together $\varphi \dashv\vdash \psi$ and $\psi \dashv\vdash \chi$ imply $\varphi \dashv\vdash \chi$.

Corollary 23:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be any formal language, $n \in \mathbb{N}$ and for any $k \in 1 \dots n$ consider some formulae φ_k and $\psi_k \in \text{form}(\mathcal{L})$ such that $\varphi_k \dashv\vdash \psi_k$. Then we also obtain

(i) $\neg\varphi_1 \;\dashv\vdash\; \neg\psi_1$

(ii) $(\varphi_1 \to \varphi_2) \;\dashv\vdash\; (\psi_1 \to \psi_2)$

(iii) If J is any composition of $\neg$ and $\to$ combining any n formulae of $\mathcal{L}$ (Nota that in fact we can build up any n-ary junctor this way), then we also get the logical equivalence of

$$J(\varphi_1, \dots, \varphi_n) \;\dashv\vdash\; J(\psi_1, \dots, \psi_n)$$

(iv) If $x \in \text{var}(\mathcal{L})$ is any variable of any $\text{sort}(x) =: i \in I$ then we also find the logical equivalences of the respective quantified formulae

$$\forall_i\, x\, \varphi_1 \;\dashv\vdash\; \forall_i\, x\, \psi_1 \;\text{ and }\; \exists_i\, x\, \varphi_1 \;\dashv\vdash\; \exists_i\, x\, \psi_1$$

Proof of 22: :
Recall that by definition (a) is $\emptyset \vdash \varphi \leftrightarrow \psi$ and again by definition this is $\emptyset \vdash (\varphi \to \psi) \wedge (\psi \to \varphi)$. Using 17.(x) this again is equivalent to (c).

Next we will prove (c) $\implies$ (d): Suppose we have $L \vdash \varphi$, then by assumption (c) and (D2) we also have $L \vdash \varphi \to \psi$. Hence we may conclude $L \vdash \psi$ by an application of (D3). The other direction that $L \vdash \psi$ implies $L \vdash \varphi$ follows by interchanging the roles of φ and ψ.

It remains to prove (d) $\implies$ (c): Look at $L := \{\varphi\}$. Then (D1) yields $L \vdash \varphi$. Hence we get $\{\varphi\} = L \vdash \psi$ by assumption (d). Therefore import-export 16 turns this into $\emptyset \vdash \varphi \to \psi$. The other part $\emptyset \vdash \psi \to \varphi$ of (c) follows by interchanging the roles of φ and ψ.

By now we have seen (a) $\iff$ (c) $\iff$ (d), next we will show (c) $\implies$ (b): As $\emptyset \vdash \varphi \to \psi$ and 17.(viii) grants $\emptyset \vdash (\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$ modus ponens allows us to proceed to $\emptyset \vdash \neg\psi \to \neg\varphi$. Likewise we find $\emptyset \vdash \neg\varphi \to \neg\psi$, so we have proved property (c) for $\neg\varphi \dashv\vdash \neg\psi$.

Direction (b) $\implies$ (c) is very similar to the last argument: By property (c) of $\neg\varphi \dashv\vdash \neg\psi$ we have $\emptyset \vdash \neg\varphi \to \neg\psi$ and $\emptyset \vdash \neg\varphi \to \neg\psi$. And by 17.(ix) we also have $\emptyset \vdash (\neg\varphi \to \neg\psi) \to (\varphi \to \psi)$ and $\emptyset \vdash (\neg\psi \to \neg\varphi) \to (\psi \to \varphi)$. So modus ponens applied to these formulae yields $\emptyset \vdash \varphi \to \psi$ and $\emptyset \vdash \psi \to \varphi$. $\square$

Proof of 23: :
Statement (i) is just property (b) in 22. So we proceed by proving statement (ii) as well. Note that by assumption we have (A) $\emptyset \vdash \varphi_2 \to \psi_2$. Also (B)

$$\{\varphi_1 \to \varphi_2\} \vdash \varphi_1 \to \varphi_2$$

is just an application of the trivial rule. Now we can connect (B) and (A) using 17.(xi) to find (C) $\{\varphi_1 \to \varphi_2\} \vdash \varphi_1 \to \psi_2$. Yet by assumption we also have (D) $\emptyset \vdash \psi_1 \to \varphi_1$ so that another dose of 17.(xi) applied to (D) and (C) yields $\{\varphi_1 \to \varphi_2\} \vdash \psi_1 \to \psi_2$. Using the import-export lemma 16 this is

$$\emptyset \vdash (\varphi_1 \to \varphi_2) \to (\psi_1 \to \psi_2)$$

As we also have $\emptyset \vdash \psi_2 \to \varphi_2$ and $\emptyset \vdash \psi_1 \to \varphi_1$ we can rerun an analogous argument to see $\emptyset \vdash (\psi_1 \to \psi_2) \to (\varphi_1 \to \varphi_2)$ and hence the logical equivalence in (ii) is established.

By induction on the generation of the junctor J out of $\neg$ and $\to$ it is clear, that statement (iii) immediately follows from (i) and (ii).

In (iv) we have $\emptyset \vdash \varphi_1 \to \psi_1$ by assumption. Hence we may use (D5) to produce $\emptyset \vdash \forall_i\, x\, (\varphi_1 \to \psi_1)$. Using this and modus ponens, we arrive at $\emptyset \vdash \varphi_1 \to \forall_i\, x\, \psi_1$. Now (L6) for $t = x$ turns this into $\emptyset \vdash \forall_i\, x\, \varphi_1 \to \forall_i\, x\, \psi_1$. Interchanging the roles of φ_1 and ψ_1 we also get $\emptyset \vdash \forall_i\, x\, \psi_1 \to \forall_i\, x\, \varphi_1$ and hence the logical equivalence.

By rule (L5) we have $\exists_i\, x\, \varphi_1 \;\dashv\vdash\; \neg\forall_i\, x\,(\neg\varphi_1)$. But by (i) and what we have just seen, we also find $\forall_i\, x\,(\neg\varphi_1) \;\dashv\vdash\; \forall_i\, x\,(\neg\psi_1)$. By (i) again this is

$$\exists_i\, x\, \varphi_1 \;\dashv\vdash\; \neg\forall_i\, x\,(\neg\varphi_1) \;\dashv\vdash\; \neg\forall_i\, x\,(\neg\psi_1) \;\dashv\vdash\; \exists_i\, x\, \psi_1$$

□

Though we do not need the following tautologies for the main goal of this article, they are very interesting in their own right. They tell us that we may commute any two universal resp. any two existential quantifiers. A mixture of these can only be commuted from $\exists\forall$ to $\forall\exists$ and not the other way around, however. While all these properties are intuitively convincing it is very reassuring to have a formal proof and not only rely on intuition.

Corollary 24:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be any formal (many-sorted, first order) language and let $\varphi \in \text{form}(\mathcal{L})$ be a formula, x and $y \in \text{var}(\mathcal{L})$ be any variable symbols of the sorts $i = \text{sort}(x)$ and $j = \text{sort}(y)$ and $t \in \text{term}(\mathcal{L})$, then we get the following tautologies and logical equivalences

(i) $\emptyset \vdash \forall_i\, x\, \forall_j\, y\, \varphi \to \forall_j\, y\, \forall_i\, x\, \varphi$

(ii) $\emptyset \vdash \exists_i\, x\, \exists_j\, y\, \varphi \to \exists_j\, y\, \exists_i\, x\, \varphi$

(iii) $\emptyset \vdash \exists_i\, x\, \forall_j\, y\, \varphi \to \forall_j\, y\, \exists_i\, x\, \varphi$

(iv) $\varphi \;\dashv\vdash\; \neg\neg\varphi$

(v) $\varphi \wedge \psi \;\dashv\vdash\; \psi \wedge \psi$

(vi) $\varphi \vee \psi \;\dashv\vdash\; \psi \vee \psi$

(vii) $\neg(\varphi \wedge \psi) \;\dashv\vdash\; (\neg\psi) \vee (\neg\psi)$

(viii) $\varphi \to \psi \;\dashv\vdash\; \neg\psi \to \neg\psi$

(ix) $\forall_i\, x\, \forall_j\, y\, \varphi \;\dashv\vdash\; \forall_j\, y\, \forall_i\, x\, \varphi$

(x) $\exists_i\, x\, \exists_j\, y\, \varphi \;\dashv\vdash\; \exists_j\, y\, \exists_i\, x\, \varphi$

(xi) $\neg(\exists_i\, x\, \varphi) \;\dashv\vdash\; \forall_i\, x\,(\neg\varphi)$

Proof:

(iv) Is proposition 17.(iii) and 17.(iv) combined with property (c) of corollary 22. Likewise (viii) here is a combination of 17.(viii) and 17.(ix) with property (c) of corollary 22.

(v) By 17.(iii) we have $\emptyset \vdash \varphi \wedge \psi \to \psi \wedge \varphi$ and interchanging the roles of φ and ψ of course $\emptyset \vdash \psi \wedge \varphi \to \varphi \wedge \psi$ as well. So by (b) in 22 we have the equivalence.

(xi) By definition of $\dashv\vdash$ rule (L5) is just $\exists_i\, x\, \varphi \;\dashv\vdash\; \neg\forall_i\, x(\neg\varphi)$. Hence by (b) in 22 we also have $\neg\exists_i\, x\, \varphi \;\dashv\vdash\; \neg\neg\forall_i\, x(\neg\varphi)$. And by (v) we have $\neg\neg\forall_i\, x(\neg\varphi) \;\dashv\vdash\; \forall_i\, x(\neg\varphi)$. Combining these logical equivalences we end up at $\neg\exists_i\, x\, \varphi \;\dashv\vdash\; \forall_i\, x(\neg\varphi)$.

(vii) By definition of the junctors $\wedge$ and $\vee$ we have $\varphi \wedge \psi = \neg(\varphi \to \neg\psi)$ and $\neg\varphi \vee \neg\psi = (\neg\neg\varphi) \to (\neg\psi)$. So by corollary 23.(i) we have a logical equivalence that by 23.(ii) and (iv) turns into

$$\neg(\varphi \wedge \psi) \;\dashv\vdash\; \varphi \to \neg\psi \;\dashv\vdash\; (\neg\neg\varphi) \to \neg\psi \;=\; (\neg\varphi) \vee (\neg\psi)$$

(vi) Let us define $\gamma := \neg\varphi$ and $\delta := \neg\psi$, then by (iv) we have $\varphi \;\dashv\vdash\; \neg\gamma$ and $\psi \;\dashv\vdash\; \neg\psi$. Then we can build up the following chain of logical equivalences

$$\begin{array}{lll} (A) & \varphi \vee \psi \;\dashv\vdash\; \neg\gamma \vee \neg\delta & \text{by 23.(iii)} \\ (B) & \neg\gamma \vee \neg\delta \;\dashv\vdash\; \neg(\gamma \wedge \delta) & \text{by (vii)} \\ (C) & \neg(\gamma \wedge \delta) \;\dashv\vdash\; \neg(\delta \wedge \gamma) & \text{by 23.(i) + (v)} \\ (D) & \neg(\delta \wedge \gamma) \;\dashv\vdash\; \neg\delta \vee \neg\gamma & \text{by (vii)} \\ (E) & \neg\delta \vee \neg\gamma \;\dashv\vdash\; \psi \vee \varphi & \text{by 23.(iii)} \end{array}$$

(i) In any formula φ the variable x is freely substitutable by x itself and $\varphi[x : x] = \varphi$. Thus we may start with (L6) twice

$$\begin{array}{lll} (A) & \emptyset \vdash \forall_j\, y\, \varphi \to \varphi & \text{by (L6)} \\ (B) & \emptyset \vdash \forall_i\, x \forall_j\, y\, \varphi \to \forall_j\, y\, \varphi & \text{by (L6)} \\ (C) & \emptyset \vdash \forall_i\, x \forall_j\, y\, \varphi \to \varphi & \text{by 17.(xi) + (B) + (A)} \\ (D) & \emptyset \vdash \varphi \to \varphi & \text{by 17.(ii)} \\ (E) & \emptyset \vdash \forall_i\, x\, (\varphi \to \varphi) & \text{by (D5) + (D)} \\ (F) & \emptyset \vdash \forall_i\, x\, (\varphi \to \varphi) \to (\varphi \to \forall_i\, x\, \varphi) & \text{by (L4)} \\ (G) & \emptyset \vdash \varphi \to \forall_i\, x\, \varphi & \text{by (D3) + (E) + (F)} \\ (H) & \emptyset \vdash \forall_j\, y\, (\varphi \to \forall_i\, x\, \varphi) & \text{by (D5) + (G)} \\ (I) & \emptyset \vdash \forall_j\, y\, (\varphi \to \forall_i\, x\, \varphi) \to (\varphi \to \forall_j\, y \forall_i\, x\, \varphi) & \text{by (L4)} \\ (J) & \emptyset \vdash \varphi \to \forall_j\, y \forall_i\, x\, \varphi & \text{by (D3) + (H) + (I)} \\ (K) & \emptyset \vdash \forall_i\, x \forall_j\, y\, \varphi \to \forall_j\, y \forall_i\, x\, \varphi & \text{by 17.(xi) + (C) + (J)} \end{array}$$

(ix) By (i) we have $\emptyset \;\vdash\; \forall_i\, x \forall_j\, y\, \varphi \to \forall_j\, y \forall_i\, x\, \varphi$ and interchanging the roles of x and y of course $\emptyset \;\vdash\; \forall_j\, y\, \varphi \forall_i\, x\, \varphi \to \forall_i\, x \forall_j\, y\, \varphi$ as well. So by (b) in 22 we have the equivalence.

(x) By (L5) applied to $\exists_j\, y\, \varphi$ we have the logical equivalence of $\exists_i\, x\, \exists_j\, y\, \varphi$ and $\neg\forall_i\, x\, (\neg\exists_j\, y\, \varphi)$. But by (xi) and 23.(iv) we also have

$$\exists_i\, x\, \exists_j\, y\, \varphi \;\dashv\vdash\; \neg\forall_i\, x\, (\neg\exists_j\, y\, \varphi) \;\dashv\vdash\; \neg\forall_i\, x \forall_j\, y\, (\neg\varphi)$$

By (ix) we may now interchange the universal quantifiers to arrive at another equivalent formula $\neg\forall_j\, y \forall_i\, x\, (\neg\varphi)$, altogether

$$\exists_i\, x\, \exists_j\, y\, \varphi \;\dashv\vdash\; \neg\forall_i\, x \forall_j\, y\, (\neg\varphi) \;\dashv\vdash\; \neg\forall_j\, y \forall_i\, x\, (\neg\varphi)$$

But we also find this formula, if we start with $\exists_j\, y\, \exists_i\, x\, \varphi$: Applying (L5) to $\exists_i\, x\, \varphi$ and another dose of (ix) we also get

$$\exists_j\, y\, \exists_i\, x\, \varphi \;\dashv\vdash\; \neg\forall_j\, y\, (\neg\exists_i\, x\, \varphi) \;\dashv\vdash\; \neg\forall_j\, y\, \forall_i\, x\, (\neg\varphi) \;\dashv\vdash\; \exists_i\, x\, \exists_j\, y\, \varphi$$

(iii) This proof will be decomposed in two parts. In the first part, we deduce $\forall_j\, y\, \varphi \to \forall_j\, y\, \exists_i\, \varphi$ as a tautology. In the second part we will deduce the claim (iii) from this.

$$\begin{array}{lll}
(A) & \emptyset \vdash \varphi \to \exists_i\, x\, \varphi & \text{by (i)} \\
(B) & \emptyset \vdash \forall_j\, y\, \big(\varphi \to \exists_i\, x\, \varphi\big) & \text{by (D5) + (A)} \\
(C) & \emptyset \vdash \forall_j\, y\, \big(\varphi \to \exists_i\, x\, \varphi\big) \to \big(\varphi \to \forall_j\, y\, \exists_i\, x\, \varphi\big) & \text{by (L4)} \\
(D) & \emptyset \vdash \varphi \to \forall_j\, y\, \exists_i\, x\, \varphi & \text{by (D3) + (C) + (D)} \\
(E) & \emptyset \vdash \forall_j\, y\, \varphi \to \varphi & \text{by (L6)} \\
(F) & \emptyset \vdash \forall_j\, y\, \varphi \to \forall_j\, y\, \exists_i\, x\, \varphi & \text{by 17.(xi) + (E) + (D)}
\end{array}$$

Now for the second part: Recall that by (L5) we have $\exists_i\, x\, \varphi \;\dashv\vdash\; \neg\forall_i\, x\, \neg\varphi$ as we then can deduce

$$\begin{array}{lll}
(G) & \emptyset \vdash \neg(\forall_j\, y\, \varphi) \to (\forall_j\, y\, \varphi) & \text{by 17.(ii)} \\
(H) & \emptyset \vdash \forall_i\, x\, \big(\neg(\forall_j\, y\, \varphi) \to \neg(\forall_j\, y\, \varphi)\big) & \text{by (D5) + (G)} \\
(I) & \emptyset \vdash \neg(\forall_j\, y\, \varphi) \to \big(\forall_i\, x\, \neg\forall_j\, y\, \varphi\big) & \text{by (D3) + (L4) + (I)} \\
(J) & \emptyset \vdash \neg\big(\forall_i\, x\, \neg\forall_j\, y\, \varphi\big) \to \forall_j\, y\, \varphi & \text{by (viii) + (iv)} \\
(K) & \emptyset \vdash \exists_i\, x\, \forall_j\, y\, \varphi \to \forall_j\, y\, \varphi & \text{by (L5)} \\
(L) & \emptyset \vdash \exists_i\, x\, \forall_j\, y\, \varphi \to \forall_j\, y\, \exists_i\, x\, \varphi & \text{by 17.(xi) + (K) + (F)}
\end{array}$$

□

Remark 25: Prenex Normal-Form:
It is good to know that any formula $\varphi \in \text{form}(\mathcal{L})$ where $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ can be turned into a logically equivalent formula in so-called *prenex normal-form*. While this form is not uniquely determined by φ it looks the following way: First of all we have $R \in \mathbb{N}$ variables $x_1, \dots, x_R \in \text{var}(\mathcal{L})$ such that $\text{sort}(x_r) = i_r \in I$ and quantifiers $Q_1, \dots, Q_R \in \text{quant}(\mathcal{L})$ such that $Q_r = \forall_{i_r}$ or $Q_r = \exists_{i_r}$. Also there is some $S \in \mathbb{N}$ and for any $s \in 1 \dots S$ also some $1 \le T(s) \in \mathbb{N}$ such that for any $s \in 1 \dots S$ and any $t \in 1 \dots T(s)$ we have a formula $\varphi_{s,t} \in \text{form}(\mathcal{L})$ such that $\varphi_{s,t}$ either is an atomic formula $\varphi_{s,t} = \alpha_{s,t} \in \text{atom}(\mathcal{L})$ or a negation thereof $\varphi_{s,t} = \neg\alpha_{s,t}$. Then φ is equivalent, to

$$\varphi \;\dashv\vdash\; Q_1 x_1 \dots Q_R x_R \bigwedge_{s=1}^{S} \left(\bigvee_{t=1}^{T(s)} \varphi_{s,t} \right)$$

While this result is really nice, its proof requires a couple of further tautologies yet as we do not need it, we abstain from proving it here.

Definition 26:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be any formal language once more and consider any set $L \subseteq \text{form}(\mathcal{L})$ of formulae therein. Then L is said to be **contradictory** iff it satisfies one of the following four equivalent properties (a) to (d). L is said to be **consistent** iff it is *not* contradictory.

(a) There is a variable symbol $x \in \text{var}(\mathcal{L})$ of any sort $i := \text{sort}(x)$ such that L allows the deduction of the formula $\neg(x =_i x)$. That is $L \vdash \neg(x =_i x)$.

(b) There is some formula $\gamma \in \text{form}(\mathcal{L})$ such that L allows the deduction of $L \vdash \gamma \wedge (\neg\gamma)$.

(c) There is some formula $\varphi \in \text{form}(\mathcal{L})$ such that L allows the deduction of both φ and $\neg\varphi$. Formally

$$L \vdash \varphi \quad \text{and} \quad L \vdash \neg\varphi$$

(d) We can deduce just any formula from L (no matter how nonsensical), that is we have the identity of

$$\text{con}(L) = \text{form}(\mathcal{L})$$

Proof:
For (a) $\Longrightarrow$ (c) we just take $\varphi := (x =_i x)$. By rule (L3) we then have $\emptyset \vdash \varphi$ and by (D2) also $L \vdash \varphi$. But we also have $L \vdash \neg\varphi$ by assumption, so this is (c). We proceed with (c) $\Longrightarrow$ (d): Choose any formula $\psi \in \text{form}(\mathcal{L})$, no matter what. Then (L2) and (D2) yield $L \vdash (\neg\varphi) \to (\varphi \to \psi)$. But as we have $L \vdash \neg\varphi$ modus ponens (D3) allows us to conclude $L \vdash \varphi \to \psi$. But we also have $L \vdash \varphi$ by assumption (c), so (D3) again yields $L \vdash \psi$. And as this has been true for any $\psi \in \text{form}(\mathcal{L})$, this is (d). In particular we find $L \vdash \gamma \wedge (\neg\gamma)$ for any formula $\gamma \in \text{form}(\mathcal{L})$. So it is clear that we also have (d) $\Longrightarrow$ (b). We now come full circle: (b) $\Longrightarrow$ (a): As we have $L \vdash \gamma \wedge (\neg\gamma)$ we also get $L \vdash \gamma$ and $L \vdash \neg\gamma$, according to 17.(xii). Now use (L2) and (D2) to write out $L \vdash (\neg\gamma) \to (\gamma \to (\neg(x =_i x))$. Then applying modus ponens (D3) twice we end up with $L \vdash \neg(x =_i x)$, which is (a). □

Remark 27:

- If $L \subseteq \text{form}(\mathcal{L})$ is consistent and we *do not* have $L \vdash \varphi$ for some formula $\varphi \in \text{form}(\mathcal{L})$, then $L \cup \{\neg\varphi\}$ is consistent, too.

  Prob Suppose $M := L \cup \{\neg\varphi\}$ was contradictory, then we had $M \vdash \varphi$ by property (c). But from $L \cup \{\neg\psi\} = M \vdash \varphi$ and $L \cup \{\varphi\} \vdash \varphi$ (which is always true, due to (D1) and (D2)) the section rule (D4) would sort out $L \vdash \varphi$, which was not the case.

- If $L \subseteq \text{form}(\mathcal{L})$ is consistent, then the set of consequences $\text{con}(L)$ of L is consistent, too. In fact we always have $\text{con}(\text{con}(L)) = \text{con}(L)$.

  Prob Let $M := \text{con}(L)$ and consider any formula φ such that $M \vdash \varphi$. This would mean we have a deduction using *finitely* many applications of the rules (L1) to (L7) (which are also present in L) and (D1) to (D5). Let μ_1 to $\mu_n \in M$ be the formluae of M involved in the proof. As $M = \text{con}(L)$ this would mean $L \vdash \mu_k$ for any $k \in 1 \dots n$. Thus we could integrate the proofs of $L \vdash \mu_k$ into the proof of $L \vdash \varphi$ already. So we have $\text{con}(M) = \text{con}(L)$. So if M was contradictory we had $\neg(x =_i x) \in \text{con}(M) = \text{con}(L)$ for some variable $x \in \text{var}(\mathcal{L})$ and therefore L would have been contradictory, to begin with.

Proposition 28:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathsf{sort})$ be any formal language once more and consider any set $L \subseteq \mathsf{form}(\mathcal{L})$ of formulae therein. Then the following three properties of L are equivalent:

(a) For any formula $\varphi \in \mathsf{form}(\mathcal{L})$ we can either deduce φ or its negation $\neg\varphi$ (but not both) from L, formally that is

$$\forall\, \varphi \in \mathsf{form}(\mathcal{L}) \;\;:\;\; L \vdash \varphi \;\;\mathsf{xor}\;\; L \vdash \neg\varphi$$

(b) For any two formulae φ, $\psi \in \mathsf{form}(\mathcal{L})$ we get the following two properties: (1) We have $L \vdash \neg\varphi$ if and only if *not* $L \vdash \varphi$ and (2) We have $L \vdash \varphi \to \psi$ if and only if $L \vdash \varphi$ implies $L \vdash \psi$.

(c) L is consistent and any set of formulae $M \subseteq \mathsf{form}(\mathcal{L})$ that allows to draw more conclusions than L already is contradictory, formally again

$$\forall\, M \subseteq \mathsf{form}(\mathcal{L}) \;\;:\;\; \mathsf{con}(L) \subset \mathsf{con}(M) \;\;\Longrightarrow\;\; M \text{ is contradictory}$$

Proof:
For (a) $\Longrightarrow$ (c) we first note that L is consistent: Choose any sort $i \in I$ and $x := x_{i,1} \in \mathsf{var}(\mathcal{L})$ and $\varphi := (x =_i x)$, then we have $\emptyset \vdash \varphi$ by (L3) and hence $L \vdash \varphi$ by (D2). Now by assumption (a) this means that we *do not* have $L \vdash \neg\varphi$ and hence property (d) of a contradictory set is ruled out.
Next we consider some $L \subseteq M \subseteq \mathsf{form}(\mathcal{L})$ such that $\mathsf{con}(L) \subset \mathsf{con}(M)$. That is there is some $\varphi \in \mathsf{con}(M)$ with $\varphi \notin \mathsf{con}(L)$. This again is $M \vdash \varphi$ and *not* $L \vdash \varphi$. By assumption on L this means $L \vdash \neg\varphi$. But as $L \subseteq M$ this also implies $M \vdash \neg\varphi$ by (D2). Hence we have found some φ with both $M \vdash \varphi$ and $M \vdash \neg\varphi$ and hence M is contradictory due to property (c).
For (c) $\Longrightarrow$ (a) we have to show that $L \vdash \varphi$ is equivalent to *not* $L \vdash \neg\varphi$. As L is consistent $L \vdash \varphi$ already implies *not* $L \vdash \neg\varphi$, as else L would satisfy property (c) of contradictory sets. It remains to show that *not* $L \vdash \varphi$ implies $L \vdash \neg\varphi$: So let us define $M := L \cup \{\neg\varphi\}$, by remark 27 above, this set is consistent, too. Hence by assumption we must have $\mathsf{con}(M) = \mathsf{con}(L)$, in particular $\neg\varphi \in M \subseteq \mathsf{con}(M) = \mathsf{con}(L)$ is $L \vdash \neg\varphi$.
For (a) $\Longrightarrow$ (b) we first consider $L \vdash \neg\varphi$ by assumption (a) we then have *not* $L \vdash \varphi$ and vice versa. Now if $L \vdash \varphi \to \psi$ and $L \vdash \varphi$, then $L \vdash \psi$ is true by modus ponens. Conversely suppose $L \vdash \varphi$ implies $L \vdash \psi$. Then we distinguish two cases: If $L \vdash \varphi$ then $L \vdash \psi$, as well and hence $\emptyset \vdash \psi \to (\varphi \to \psi$ combined with modus ponens yields $L \vdash \varphi \to \psi$. If *not* $L \vdash \varphi$ then we have $L \vdash \neg\varphi$ by (a). Thus $\emptyset \vdash (\neg\varphi) \to (\varphi \to \psi)$ and modus ponens yields $L \vdash \varphi \to \psi$ in this case, too.
For (b) $\Longrightarrow$ (a) we first consider the case $L \vdash \neg\varphi$. By assumption we then have *not* $L \vdash \varphi$ and hence (a). In the other case *not* $L \vdash \neg\varphi$ we have $L \vdash \neg\neg\varphi$ by assumption. But 17.(iv) $\emptyset \vdash \neg\neg\varphi \to \varphi$ would turn this into $L \vdash \varphi$, too. Hence (a) is satisfied in both cases. $\Box$

6 Defining Many-Sorted Models

We will now introduce the notion of a realization of a formal language, which will lead us to the definition of models. Roughly speaking, a realization of a given language is a mathematical object in which all the symbols of the language can be interpreted. A model of a given list of axioms is a realization of the language (the axioms are formulated in), in which all the axioms come true. E.g. a group is a model of the group axioms (associativity, neutral element and inverse element). The pair (R, M) of a module M over a commutative ring R is a model for the axioms of modules over a commutative rings, as presented in example 9.
The question arises naturally what a mathematical object in this sense is. The answer of course has to be a (family of) set(s), as in the end all that mathematics talks about is sets. In turn leads to the question what sets are. Having established formal languages one could already take all the following sets to be understood in formal set theory (i.e. variable symbols of a second formal language). However, we've already started on the assumption that the reader understands naive set-theory, so there is no reason to think it has been forgotten by now. Thus the sets used in the following are best understood naively again.

Definition 29:
Let $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a many-sorted (first order) language, then we define the following notions:

(i) **$\mathcal{L}$-structure:** We call the ordered pair (X, ϱ) a **realization of $\mathcal{L}$** or **$\mathcal{L}$-structure** iff $X = (X_i)$ is an I-tuple of non-empty sets and ϱ is a function satisfying the following properties:

- for any constant symbol $c \in \text{const}(\mathcal{L})$ of $\text{sort}(c) = i$, the realization $\varrho(c)$ is an element in X_i

$$\varrho(c) \quad \in \quad X_i$$

- for any function symbol $f \in \text{func}(\mathcal{L})$ of $\text{sort}(f) = (i_1, \dots, i_k, i_{k+1})$, the realization $\varrho(f)$ of f is a function of the following form

$$\varrho(f) \; : \; X_{i_1} \times \cdots \times X_{i_k} \to X_{i_{k+1}}$$

- for any relation symbol $R \in \text{rel}(\mathcal{L})$ of $\text{sort}(R) = (i_1, \dots, i_k)$, the realization $\varrho(R)$ of R is a relation of the following form

$$\varrho(R) \quad \subseteq \quad X_{i_1} \times \cdots \times X_{i_k}$$

And we denote the class (note that this is not a set) of all realizations of $\mathcal{L}$ by:

$$\text{real}(\mathcal{L}) \quad := \quad \{ (X, \varrho) \mid (X, \varrho) \text{ is an } \mathcal{L}\text{-structure} \}$$

(ii) **Assignments:** Let (X, ϱ) be a realization of $\mathcal{L}$, then ω is said to be an **assignment** of (X, ϱ), iff it is a mapping of the form

$$\omega \; : \; \text{var}(\mathcal{L}) \to \bigcup_{i \in I} X_i \qquad \text{satisfying}$$

$$\forall\, x \in \text{var}(\mathcal{L}) \quad \text{we get} \quad \text{sort}(x) = i \implies \omega(x) \in X_i$$

And we denote $\mathsf{ass}(X,\varrho) := \{\,\omega \mid \omega \text{ is an assignment of } (X,\varrho)\,\}$ the set of assignments of a fixed realization (X,ϱ). If we are now given a variable symbol $x \in \mathsf{var}(\mathcal{L})$ of the sort $i := \mathsf{sort}(x)$, an element $a \in X_i$ and an assignment $\omega \in \mathsf{ass}(X,\varrho)$, then we define another assignment $\omega[x:a]$ by

$$\omega[x:a] \;:\; \mathsf{var}(\mathcal{L}) \to \bigcup_{i\in I} X_i \;:\; y \mapsto \begin{cases} \omega(y) & \text{for } y \neq x \\ a & \text{for } y = x \end{cases}$$

(iii) **Evaluation of terms:** If again (X,ϱ) is a realization of $\mathcal{L}$ and $\omega \in \mathsf{ass}(X,\varrho)$ is an assignment, then we define the function $\mathsf{val}^{\omega}_{(X,\varrho)} : \mathsf{term}(\mathcal{L}) \to \bigcup_{i\in I} X_i$ by recursion on the generation of terms

$$\begin{aligned} \mathsf{val}^{\omega}_{(X,\varrho)}(x) &:= \omega(x) \\ \mathsf{val}^{\omega}_{(X,\varrho)}(c) &:= \varrho(c) \\ \mathsf{val}^{\omega}_{(X,\varrho)}(f t_1 \dots t_k) &:= \varrho(f)(\mathsf{val}^{\omega}_{(X,\varrho)}(t_1), \dots, \mathsf{val}^{\omega}_{(X,\varrho)}(t_k)) \end{aligned}$$

(iv) **evaluation of formulae**
Further we define another function (denoted by the same name) $\mathsf{val}^{\omega}_{(X,\varrho)} : \mathsf{form}(\mathcal{L}) \to \mathbb{B}$ by recursion on the generation of formulae

$$\begin{aligned} \mathsf{val}^{\omega}_{(X,\varrho)}(t_1 =_i t_2) &:= \begin{cases} 1 & \text{if } \mathsf{val}^{\omega}_{(X,\varrho)}(t_1) = \mathsf{val}^{\omega}_{(X,\varrho)}(t_2) \\ 0 & \text{else} \end{cases} \\ \mathsf{val}^{\omega}_{(X,\varrho)}(R t_1 \dots t_k) &:= \begin{cases} 1 & \text{if } (\mathsf{val}^{\omega}_{(X,\varrho)}(t_1), \dots, \mathsf{val}^{\omega}_{(X,\varrho)}(t_k)) \in \varrho(R) \\ 0 & \text{else} \end{cases} \\ \mathsf{val}^{\omega}_{(X,\varrho)}(\neg\varphi) &:= 1 - \mathsf{val}^{\omega}_{(X,\varrho)}(\varphi) \\ \mathsf{val}^{\omega}_{(X,\varrho)}(\varphi \to \psi) &:= \max\Big\{\, 1 - \mathsf{val}^{\omega}_{(X,\varrho)}(\varphi), \mathsf{val}^{\omega}_{(X,\varrho)}(\psi) \,\Big\} \\ \mathsf{val}^{\omega}_{(X,\varrho)}(\forall_i\, x\, \varphi) &:= \begin{cases} 1 & \text{if } X_i = \{a \in X_i \mid \mathsf{val}^{\omega[x:a]}_{(X,\varrho)}(\varphi) = 1\} \\ 0 & \text{else} \end{cases} \\ \mathsf{val}^{\omega}_{(X,\varrho)}(\exists_i\, x\, \varphi) &:= \begin{cases} 1 & \text{if } \emptyset \neq \{a \in X \mid \mathsf{val}^{\omega[x:a]}_{(X,\varrho)}(\varphi) = 1\} \\ 0 & \text{else} \end{cases} \end{aligned}$$

Nota here we used $\mathbb{B} := \{\,0,1\,\}$ and $0 \in \mathbb{B}$ represents the logical value *false*, whereas $1 \in \mathbb{B}$ is to be interpreted, as *true*. If we identify $\mathbb{B} = \mathbb{Z}_2$, then $+$ becomes the *xor* junctor and $\cdot$ the *and* junctor. In this sense $\mathsf{val}^{\omega}_{(X,\varrho)}(\varphi)$ carries the logical value of $\mathsf{val}^{\omega}_{(X,\varrho)}(\varphi) = 1$, which is either *true* or *false*. You might also want to consult remark 34 for some basic comments on this.

Lemma 30: Coincidence Theorem:
Let $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathsf{sort})$ be a many-sorted (first order) language, (X,ϱ) be any realization of $\mathcal{L}$ and consider any two assignments $\alpha, \beta \in \mathsf{ass}(X,\varrho)$. Finally let $t \in \mathsf{term}(\mathcal{L})$ be a term and $\varphi \in \mathsf{form}(\mathcal{L})$ be a formula of $\mathcal{L}$. Then the evaluation of t and φ only depends on the evaluation at the free variables, that is

$$\begin{aligned} \forall\, x \in \mathsf{free}(t) \,:\, \alpha(x) = \beta(x) &\implies \mathsf{val}^{\alpha}_{(X,\varrho)}(t) = \mathsf{val}^{\beta}_{(X,\varrho)}(t) \\ \forall\, x \in \mathsf{free}(\varphi) \,:\, \alpha(x) = \beta(x) &\implies \mathsf{val}^{\alpha}_{(X,\varrho)}(\varphi) = \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi) \end{aligned}$$

In particular: If φ is a sentence of $\mathcal{L}$ (that is $\text{free}(\varphi) = \emptyset$) then the truth of φ is fixed. That is $\text{val}^{\omega}_{(X,\varrho)}(\varphi)$ is identical for all assignments $\omega \in \text{ass}(X, \varrho)$.

Proposition 31:
As always let $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a many-sorted language, $x \in \text{var}(\mathcal{L})$ be a variable symbol of the $\text{sort}(x) =: i \in I$, s, $t \in \text{term}(\mathcal{L})$ be terms and $\varphi \in \text{form}(\mathcal{L})$ be a formula of $\mathcal{L}$. Also let (X, ϱ) be a realization of $\mathcal{L}$ and $\omega \in \text{ass}(X, \varrho)$ be an assignment of (X, ϱ). Then the evaluation of $s[x : t]$ with ω is just the same as the evaluation of s with $\omega[x : b]$ where $b := \text{val}^{\omega}_{(X,\varrho)}(t)$

$$\text{val}^{\omega}_{(X,\varrho)}\big(s[x : t]\big) \;\;=\;\; \text{val}^{\omega[x:b]}_{(X,\varrho)}(s)$$

And if x is freely substitutable by t in φ, then it is also true that the evaluation of $\varphi[s : t]$ with ω is the same as the evaluation of φ with $\omega[x : b]$

$$\text{val}^{\omega}_{(X,\varrho)}\big(\varphi[x : t]\big) \;\;=\;\; \text{val}^{\omega[x:b]}_{(X,\varrho)}(\varphi)$$

Notation 32:
We will often fix a realization (X, ϱ) of $\mathcal{L}$ and then all statements that are based on a realization are understood to refer to it. In this case we will abbreviate our notation in the following way

(i) Let $\varphi \in \text{form}(\mathcal{L})$ be a formula and $t \in \mathcal{L}$ be a term of $\mathcal{L}$, if we are now given an assignment $\omega \in \text{ass}(X, \varrho)$ of (X, ϱ), then we simply write

$$\begin{aligned} t(\omega) &:= \text{val}^{\omega}_{(X,\varrho)}(t) \qquad \in X_{\text{sort}(t)} \\ \varphi(\omega) &:= \text{val}^{\omega}_{(X,\varrho)}(\varphi) = 1 \quad \text{(true or false)} \end{aligned}$$

(ii) Let $\varphi(x_1, \dots, x_k) \in \text{form}(\mathcal{L})$ be a formula and $t(x_1, \dots, x_k) \in \text{term}(\mathcal{L})$ be a term, both having the free variables $\text{free}(\varphi) = \{x_1, \dots, x_k\} = \text{free}(t)$. If we are now given elements $a_j \in X_{\text{sort}(x_j)}$ (where $j \in 1 \dots k$), then we solely write

$$\begin{aligned} t(a_1, \dots, a_k) &:= \text{val}^{\omega[x_1:a_1,\dots,x_k:a_k]}_{(X,\varrho)}(t) \qquad \in X_{\text{sort}(t)} \\ \varphi(a_1, \dots, a_k) &:= \text{val}^{\omega[x_1:a_1,\dots,x_k:a_k]}_{(X,\varrho)}(\varphi) = 1 \quad \text{(true or false)} \end{aligned}$$

Nota that by lemma 30 the value of $t(a_1, \dots, a_k)$ and $\varphi(a_1, \dots, a_k)$ respectively is independent of the choice of the assignment $\omega \in \text{ass}(X, \varrho)$, since all free variables are covered.

Proof: of Lemma 30
The proof is conducted by induction on the number of steps used to build up t or φ respectively. Let us start with the term t: If $t = c$ for some constant symbol c the evaluation is done by ϱ not by α or β, so in this case there is nothing to prove, really. And if $t = x$ for some variable symbol x then $x \in \text{free}(t)$, as terms never forget free variables. So by assumption we have

$$\text{val}^{\alpha}_{(X,\varrho)}(c) \;=\; \varrho(c) \;=\; \text{val}^{\beta}_{(X,\varrho)}(t)$$

$$\mathsf{val}^{\alpha}_{(X,\varrho)}(x) \ = \ \alpha(x) \ = \ \beta(x) \ = \ \mathsf{val}^{\beta}_{(X,\varrho)}(x)$$

In case $t = ft_1 \dots t_k$ for a function symbol f and terms $t_i \in \mathsf{term}(\mathcal{L})$ we have $\mathsf{free}(t_i) \subseteq \mathsf{free}(t)$ and hence by induction hypothesis $\mathsf{val}^{\alpha}_{(X,\varrho)}(t_i) = \mathsf{val}^{\beta}_{(X,\varrho)}(t_i)$. And thereby we get the induction step

$$\begin{aligned}
\mathsf{val}^{\alpha}_{(X,\varrho)}(t) &= \varrho(f)\big(\mathsf{val}^{\alpha}_{(X,\varrho)}(t_1), \dots, \mathsf{val}^{\alpha}_{(X,\varrho)}(t_k)\big) \\
&= \varrho(f)\big(\mathsf{val}^{\beta}_{(X,\varrho)}(t_1), \dots, \mathsf{val}^{\beta}_{(X,\varrho)}(t_k)\big) \ = \ \mathsf{val}^{\beta}_{(X,\varrho)}(t)
\end{aligned}$$

So we turn our attention to the formula φ: If $\varphi = (s =_i)$ for some terms s, t then $\mathsf{free}(\varphi) = \mathsf{free}(s) \cup \mathsf{free}(t)$ and hence we have $\mathsf{val}^{\alpha}_{(X,\varrho)}(s) = \mathsf{val}^{\beta}_{(X,\varrho)}(s)$ and $\mathsf{val}^{\alpha}_{(X,\varrho)}(t) = \mathsf{val}^{\beta}_{(X,\varrho)}(t)$ by what we have just proved. So whether these two values are equal or not does not depend on α or β and hence we have $\mathsf{val}^{\alpha}_{(X,\varrho)}(\varphi) = \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi)$ in this case. The same is true for $\varphi = Rt_1 \dots t_k$ for some relation symbol R and terms $t_i \in \mathsf{term}(\mathcal{L})$. We find $\mathsf{free}(t_i) \subseteq \mathsf{free}(\varphi)$ and hence α and β agree on whether the k-tuple is contained in $\varrho(R)$, or not

$$\big(\mathsf{val}^{\alpha}_{(X,\varrho)}(t_i), \dots, \mathsf{val}^{\alpha}_{(X,\varrho)}(t_i)\big) \ = \ \big(\mathsf{val}^{\beta}_{(X,\varrho)}(t_i), \dots, \mathsf{val}^{\beta}_{(X,\varrho)}(t_i)\big)$$

Next we consider a formula of the form $\neg\varphi$. As $\mathsf{free}(\neg\varphi) = \mathsf{free}(\varphi)$ we can use the induction hypothesis to find the equality of

$$\mathsf{val}^{\alpha}_{(X,\varrho)}(\neg\varphi) \ = \ 1 - \mathsf{val}^{\alpha}_{(X,\varrho)}(\varphi) \ = \ 1 - \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi) \ = \ \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi)$$

Likewise for a formula of the form $\varphi \to \psi$ where we have $\mathsf{free}(\varphi)$ and $\mathsf{free}(\psi) \subseteq \mathsf{free}(\varphi \to \psi)$ and hence can compute

$$\begin{aligned}
\mathsf{val}^{\alpha}_{(X,\varrho)}(\varphi \to \psi) &= \max\Big\{ 1 - \mathsf{val}^{\alpha}_{(X,\varrho)}(\varphi), \mathsf{val}^{\alpha}_{(X,\varrho)}(\varphi) \Big\} \\
&= \max\Big\{ 1 - \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi), \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi) \Big\} \ = \ \mathsf{val}^{\beta}_{(X,\varrho)}(\varphi \to \psi)
\end{aligned}$$

Finally for rule (F3) we consider a formula of the form $\forall_i\, x\varphi$ or $\exists_i\, x\varphi$. In this case x no longer is a free variable, so we can only use the induction hypothesis on $\mathsf{free}(\varphi) \setminus \{ x \}$. But $\alpha[x : a](x) = x = \beta[x : a](x)$ and hence we have $\alpha[x : a](y) = x = \beta[x : a](y)$ for any $y \in \mathsf{free}(\varphi)$ again. Hence the evaluation of φ by $\alpha[x : a]$ and $\beta[x : a]$ yields the same result, by induction hypothesis and therefore we have the equality of sets

$$\Big\{ a \in X_i \ \Big| \ \mathsf{val}^{\alpha[x:a]}_{(X,\varrho)}(\varphi) \Big\} \ = \ \Big\{ a \in X_i \ \Big| \ \mathsf{val}^{\beta[x:a]}_{(X,\varrho)}(\varphi) \Big\}$$

And as these sets are equal, it does not matter if we determine, whether they are are equal to X_i or not (for $\forall_i\, x\varphi$) or non-empty (for $\exists_i\, x\varphi$) or not based on α or β. As these have been all the rules how to build up a formula, our induction s complete. □

Proof: of Proposition 31:
Of course we have to prove this by induction on the number of steps used to build up the term s or formula φ respectively. We start with the term s:

- If $s = c$ is a constant symbol of $\mathcal{L}$, then $s[x : t] = c$ again and hence $s[x : t](\omega) = c(\omega) = \varrho(c)$ and $s(\omega[x : b]) = c(\omega[x : b]) = \varrho(c)$, as well.

- If $s = y$ is a variably symbol of $\mathcal{L}$ then we have to distinguish two cases: (1) If $y \neq x$ then $s[x:t] = y$ again and hence $s[x:t](\omega) = y(\omega) = \omega(y)$. But likewise $s(\omega[x:b]) = \omega[x:b](y) = \omega(y)$ such that $s[x:t](\omega) = s(\omega[x:b])$. (2) If $y = x$ then we get $s[x:t] = t$ such that $s[x:t](\omega) = \omega(t) = b$. Alas we also find $s(\omega[x:b]) = \omega[x:b](x) = b$ and hence $s[x:t](\omega) = s(\omega[x:b])$ again.

- If $s = ft_1 \dots t_k$ where f is a function symbol and the t_j are terms of $\mathcal{L}$ we have $t_j[x:t](\omega) = t_j(\omega[x:b])$ by assumption. Then the induction hypothesis can be used in the substitution

$$\begin{aligned}\big(ft_1 \dots t_k\big)[x:t](\omega) &= \big(ft_1[x:t] \dots t_k[x:t]\big)(\omega) \\ &= \varrho(f)\big(t_1[x:t](\omega), \dots, t_k[x:t](\omega)\big) \\ &= \varrho(f)\big(t_1(\omega[x:b]), \dots, t_k(\omega[x:b])\big) \\ &= \big(ft_1 \dots t_k\big)(\omega[x:b])\end{aligned}$$

As these are all rules how to build up a term, we have proved the claim for the terms like s. Now for the formula φ we continue with our induction

- The case where $\varphi = Rt_1 \dots t_k$ is similar to the one before. For ease of notation we identify 0 with *false* and 1 with *true*, then we get

$$\begin{aligned}\big(Rt_1 \dots t_k\big)[x:t](\omega) &= \big(Rt_1[x:t] \dots t_k[x:t]\big)(\omega) \\ &= \big(t_1[x:t](\omega), \dots, t_k[x:t](\omega)\big) \in \varrho(R) \\ &= \big(t_1(\omega[x:b]), \dots, t_k(\omega[x:b])\big) \in \varrho(R) \\ &= \big(Rt_1 \dots t_k\big)(\omega[x:b])\end{aligned}$$

- If $\varphi = (t_1 =_i t_2)$ for some terms t_1 and t_2 we have already seen the equality $t_j[x:t](\omega) = t_j(\omega[x:b])$ and hence

$$\begin{aligned}(t_1 =_i t_2)[x:t](\omega) &= (t_1[x:t](\omega) = t_2[x:t](\omega)) \\ &= (t_1(\omega[x:b]) = t_2(\omega[x:b])) \\ &= (t_1 =_i t_2)(\omega[x:b])\end{aligned}$$

- If $\varphi = \neg\psi$ for some formula ψ, then x is freely substitutable by t in ψ, as well. So by induction hypothesis $\psi[x:t](\omega) = \psi(\omega[x:b])$, thus

$$\varphi[x:t](\omega) \;=\; 1 - \psi[x:t](\omega] \;=\; 1 - \psi(\omega[x:b]) \;=\; \varphi(\omega[x:b])$$

  Precisely the same argumentation can be applied to $\varphi = (\psi_1 \to \psi_2)$ for some formulae ψ_1 and ψ_2 of $\mathcal{L}$. In this case we compute

$$\begin{aligned}\big(\psi_1 \to \psi_2\big)[x:t](\omega) &= \max\{\, 1 - \psi_1[x:t](\omega), \psi_2[x:t](\omega) \,\} \\ &= \max\{\, 1 - \psi_1(\omega[x:b]), \psi_2(\omega[x:b]) \,\} \\ &= \big(\psi_1 \to \psi_2\big)(\omega[x:b])\end{aligned}$$

- Finally, if $\varphi = \forall_j\, y\, \psi$ for some variable symbol y of the $j := \mathrm{sort}(y)$ and some formula ψ then we have to distinguish two cases: (1) if $y = x$, that is $\varphi = \forall_i\, x\, \psi$, we have $\varphi[x : t] = \varphi$ and hence

$$\varphi[x : t](\omega) \;=\; \varphi(\omega) \;=\; \big(\{\, a \in X_i \mid \psi(\omega[x : a]) = 1 \,\} = X_i\big)$$

where we identified $0 = $ *false* and $1 = $ *true* again. On the other hand we can compute what happens if we evaluate φ under $\omega[x : b]$

$$\varphi(\omega[x : b]) \;=\; \big(\{\, a \in X_i \mid \psi(\omega[x : b][x : a]) = 1 \,\} = X_i\big)$$

As $\omega[x : b][x : a] = \omega[x : a]$ we see that $\varphi[x : t](\omega) = \varphi(\omega[x : b])$ again. It remains the second case (2) $y \neq x$: In this case we have $\varphi[x : t] = \forall_j\, y\, \psi[x : t]$, so

$$\varphi[x : t](\omega) \;=\; \big(\{\, c \in X_j \mid \psi[x : t](\omega[y : c]) = 1 \,\} = X_j\big)$$

By induction hypothesis we have $\psi[x : t](\omega[y : c]) = \psi(\omega[y : c][x : b'])$ where $b' := \omega[y : c](t)$. By assumption x was freely substitutable by t in φ and φ contained a formula of the form $\forall_j\, y\, \psi \leq \varphi$, so this implies $y \notin \mathrm{free}(t)$ and hence $b' = \omega[y : c](t) = \omega(t) = b$ by the coincidence lemma 30. Therefore we get

$$\varphi[x : t](\omega) \;=\; \big(\{\, c \in X_j \mid \psi(\omega[y : c][x : b]) = 1 \,\} = X_j\big)$$

Again we compare this to the evaluation of φ by $\omega[x : b]$. In this case

$$\varphi(\omega[x : b]) \;=\; \big(\{\, c \in X_j \mid \psi(\omega[x : b][y : c]) = 1 \,\} = X_j\big)$$

But as $x \neq y$ we have $\omega[y : c][x : b] = [x : b][y : c]$ and therefore $\varphi[x : t](\omega) = \varphi(\omega[x : b])$. The formula $\varphi = \exists_i\, x\, \psi$ can be handled in complete analogy - all we have to do is replace the condition "$= X_j$" by "$\neq \emptyset$". So altoghether we have completed the induction.

$\square$

Definition 33:
Let $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$ be a (many-sorted, first-order) formal language and fix $(X, \varrho) \in \mathrm{real}(\mathcal{L})$ a realization of $\mathcal{L}$. Then we define the following notions

(i) Given $k \in \mathbb{N}$ and $i_1, \dots, i_k \in I$, we say that $D \subseteq X_{i_1} \times \cdots \times X_{i_k}$ is **defined by** the formula $\varphi \in \mathrm{form}(\mathcal{L})$, iff $\mathrm{free}(\varphi) = \{\, x_1, \dots, x_k \,\}$ such that for any $j \in 1 \dots k$ we have $\mathrm{sort}(x_j) = i_j$ and

$$D \;=\; \{\, (a_1, \dots, a_k) \in X_{i_1} \times \cdots \times X_{i_k} \mid \varphi(a_1, \dots, a_k) = 1 \,\}$$

And D is called **definable** iff there is some formula $\varphi \in \mathrm{form}(\mathcal{L})$ such that D is definable by φ.

(ii) Now a function $f : X_{i_1} \times \cdots \times X_{i_k} \to X_{i_{k+1}}$ is said to be **definable** iff its graph is such - where the graph of f is defined to be the following subset of $X_{i_1} \times \cdots \times X_{i_{k+1}}$

$$\mathrm{graph}\,(f) \;:=\; \Big\{\, (a_1, \dots, a_k, f(a_1, \dots, a_k)) \mid a_j \in X_{i_j}\, (j \in 1 \dots n) \,\Big\}$$

(iii) Let now $i_1, \dots, i_k, i_{k+1} \in I$ and consider any function $f : X_{i_1} \times \dots \times X_{i_k} \to X_{i_{k+1}}$, then the equation

$$f(a_1, \dots, a_k) \;=\; a_{k+1}$$

is said to be **equivalent** to the formula $\varphi \in \mathrm{form}(\mathcal{L})$ iff the graph of f is defined by φ. And explicitly this reads as $\mathrm{free}(\varphi) = \{ x_1, \dots, x_k, x_{k+1} \}$ where $\mathrm{sort}(x_j) = i_j$ for any $j \in 1 \dots k+1$ and

$$f(a_1, \dots, a_k) \;=\; a_{k+1} \quad \Longleftrightarrow \quad \varphi(a_1, \dots, a_k, a_{k+1}) = 1$$

Remark 34: Some Junctorial Logic:
Let us enlighten the evaluation of formulae a bit and recall some basic notions of **junctorial logic**: There are two logical values: *true* and *false* (even fuzzy logic formulates statements that are either right or wrong, like "The truthfulness of φ exceeds 0.5"). So it makes sense to represent the logical values as numbers - 0 for *false* and 1 for *true*. So the set $\mathbb{B} := \{ 0, 1 \}$ represents these logical values. In this sense the truthfulness of a formula $\varphi \in \mathrm{form}(\mathcal{L})$ is a number, 0 or 1.
A n-ary **junctor** J connects n formulae $\varphi_1, \dots, \varphi_n$ to form a new formula $J(\varphi_1, \dots, \varphi_n)$. Two of the most prominent examples are "φ_1 and φ_2" and "if φ_1 then φ_2". The truth of the statement $J(\varphi_1, \dots, \varphi_n)$ should only depend on the truthfulness of the ingredients $\varphi_1, \dots, \varphi_n$. Such a junctor is called *extensional*, in contrast to *intentional* junctors that depend on the formulation of $\varphi_1, \dots, \varphi_n$. Sadly most human beings are intentional junctors, but mathematics restricts to extensional junctors.
So in the end a n-ary extensional junctor J gives rise to a function that assigns the n logical values of $\varphi_1, \dots, \varphi_n$ the logical value of the composition $J(\varphi_1, \dots, \varphi_n)$. We denote this fuction by J again, as $J : \mathbb{B}^n \to \mathbb{B}$ fully describes the logical impact of the junctor J. As an example let us present a list of the most commonly used extensional junctors here:

A	B	not A	A or B	A and B	A xor B	if A then B	A iff B
0	0	1	0	0	0	1	1
0	1	1	1	0	1	1	0
1	0	0	1	0	1	0	0
1	1	0	1	1	0	1	1

Clearly we write $\neg\varphi$ for "not φ" and "$A \implies B$" for "if A then B" all the time. All the other junctors have symbols, too: "A iff B" is "$A \iff B$" and we also interpret "$A \wedge B$" as "A and B", resp. "$A \vee B$" as "A or B". If we equip $\mathbb{B}$ with the operations $+$ and $\cdot$ of $\mathbb{Z}_2$ then we can also compute the value of these juctors:

$$\begin{aligned}
\text{not } A &= 1 - A \\
A \text{ and } B &= A \cdot B \\
A \text{ xor } B &= A + B \\
A \text{ iff } B &= 1 - (A + B) \\
A \text{ or } B &= \max\{ A, B \} \;=\; 1 - (1 - A) \cdot (1 - B) \\
\text{if } A \text{ then } B &= \max\{ 1 - A, B \} \;=\; 1 - A \cdot (1 - B)
\end{aligned}$$

So these have been precisely the operations by which we defined the evaluation of the formulae $\neg\varphi$ and $\varphi \to \psi$ under an assignment ω. In doing this we already implemented that $\neg$ and $\to$ are extensional, not intentional. Seeing this it is not too surprising that *any* extensional junctor can be expressed as a combination of $\neg$ and $\to$. Likewise as a combination of $\neg$ and $\wedge$. In this sense we can import any extensional junctor into $\mathcal{L}$, as we did before, with $\wedge$, $\vee$ and $\leftrightarrow$.
Now that we know what it means to evaluate a formula, we wish to use this notion to formalize what it means that a family L of formulae *implies* another formula φ. In this context to imply means: Presuming you agree that all the formula of L are true, you also have to agree that φ is true. Thus the question of implication is very closely related to the the question of what a mathematical proof truly is. There are two immediately sensible concepts of what a mathematical proof is:

1. For the one part one could fix a list of sentences and agree that these are considered to be true in any case. An obvious candidate for this is the formula $x =_i x$. This is certainly true no matter what content x takes. Then one could fix a list of rules telling which deductions we consider to be evident. The most reasonable example for such a rule is the following: If both $\varphi \to \psi$ and φ are already agreed to be true, then we agree that ψ is true as well. This is precisely the path we took in section 4. The proof of the uniqueness of the inverse element of $x \in G$ in a group G is an example of such a deduction: If x' and x'' have the properties of an inverse element of x, then we can deduce $x' = x''$ by applying the group axioms: $x' = x'1 = x'(xx'') = (x'x)x'' = 1x'' = x''$.

2. On the other hand one could come up with the idea of "testing" whether a formula holds true - that is, we choose some realization (X, ϱ) and some assignment ω and try whether $\varphi(\omega)$ is true. If this is the case for any realization and any assignment, then certainly φ is to be considered "true" in some sense. The proof of the degree formula $[F : D] = [F : E] \cdot [E : F]$ of field extensions is an example of such an approach: Given any field extensions $F : E$ and $E : D$ we choose bases (y_k) of F as a E-vector space and (x_i) of E as a D-vector space. Then it is shown, that $(x_i y_k)$ is a basis of F as a D-vector space. So in this proof we used the specific properties of the field extensions, other fields would have had bases of different size!

Both approaches to proofs are convincing in their own right. And even though the first method seems to be more straightforward, we find that the second method is more common in mathematical practice. So the question arises whether these two notions of a proof really have different consequences. Is there a formula that can be proved in one way, but not by the other? Could it even happen that one way allows to prove φ and the other $\neg\varphi$?
As both kinds of arguments are commonplace in mathematical proofs it would be extremely irritating if there was a deviation between these two notions. And in fact in 1929 Kurt Gödel succeeded in proving that these two notions truly are equivalent (viz. [4]). We should note that K. Gödel proved this result (the so-called Correctness and Completeness Theorems) for one-sorted logic. Yet another proof makes use of Henkin's Lemma (viz. [5]) and it is this proof that we will transfer to the many-sorted case.
In fact it also is possible to reduce any many-sorted language with finitely many sorts to a one-sorted logic that also includes predicates to represent the different sorts. Using this method we could prove that the Correctness and Completeness theorems are inherited by the many-sorted language from the one-sorted language. Yet while this is possible, it takes more technical effort, so we deem it best to start with many-sorted languages right away and only take one-sorted languages as the special case they are.

Definition 35:
Let $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a many-sorted, first order language again. Further let (X, ϱ) be a realization of $\mathcal{L}$ and $\omega \in \text{ass}(X, \varrho)$ be an assignment of (X, ϱ). If now $\varphi \in \text{form}(\mathcal{L})$ is a single and $L \subseteq \text{form}(\mathcal{L})$ is a set of formulae of $\mathcal{L}$, then we introduce the following symbols

- We say that φ **holds true under** ω **in** (X, ϱ) iff the evaluation of φ under ω is true, we write

$$(X, \varrho) \models_\omega \varphi \quad :\Longleftrightarrow \quad \text{val}^\omega_{(X,\varrho)}(\varphi) = 1$$

- We say that φ is **universally true** in (X, ϱ) iff φ holds true under any assignment of (X, ϱ). And we say that (X, ϱ) is a **model** of L iff any formula of L is universally true in (X, ϱ), we write

$$\begin{aligned} (X, \varrho) \ \models\ \varphi \quad &:\Longleftrightarrow \quad \forall\, \omega \in \text{ass}(X, \varrho) \text{ we get } (X, \varrho) \models_\omega \varphi \\ (X, \varrho) \ \models\ L \quad &:\Longleftrightarrow \quad \forall\, \lambda \in L \text{ we get } (X, \varrho) \ \models\ \lambda \end{aligned}$$

  We write $\text{con}(X, \varrho) := \{\, \varphi \in \text{form}(\mathcal{L}) \mid (X, \varrho) \ \models\ \varphi \,\}$ for the set of all formula that are universally true in (X, ϱ) and $\text{model}(L) := \{\, (X, \varrho) \in \text{real}(\mathcal{L}) \mid (X, \varrho) \ \models\ L \,\}$ for the class of all models of L.

- And finally we say that L **implies** φ **in** (X, ϱ) iff any assignment of (X, ϱ), that satisfies all the formulae of L also satisfies φ. And we simply say that L **implies** φ iff L implies φ in any realization (X, ϱ). Formally we write

$$\begin{aligned} (X, \varrho) \models_L \varphi \quad &:\Longleftrightarrow \quad \left\{ \begin{array}{c} \forall\, \omega \in \text{ass}(X, \varrho) \text{ we get} \\ (\forall\, \lambda \in L : (X, \varrho) \models_\omega \lambda) \implies (X, \varrho) \models_\omega \varphi) \end{array} \right. \\ L \ \models\ \varphi \quad &:\Longleftrightarrow \quad \forall\, (X, \varrho) \in \text{real}(\mathcal{L}) \text{ we get } (X, \varrho) \models_L \varphi \end{aligned}$$

So let us summarize what we just defined here: We say $L \models \varphi$ iff for any realization (X, ϱ) of $\mathcal{L}$ (that is in any structure in which we can talk in the language $\mathcal{L}$) and any assignment $\omega \in \text{ass}(X, \varrho)$ (that is under any possible interpretation of the symbols of $\mathcal{L}$ in (X, ϱ)), that whenever all the assumptions $\lambda \in L$ are satisfied then the conclusion φ is satisfied, as well. There is not a single exception to this. Therefore the notion $L \models \varphi$ is the correct formalization of φ *can be proved under the assumptions* L.

Nota if $\varphi \in \text{sen}(\mathcal{L})$ does not contain free variables, then by 30 the assignment ω doesn't really matter, that is we have $\exists\, \omega \in \text{ass}(X, \varrho) : (X, \varrho) \models_\omega \varphi$ if and only if $(X, \varrho) \ \models\ \varphi$. And for any $\varphi \in \text{form}(\mathcal{L})$ it is easy to see, that

$$(X, \varrho) \ \models\ L \text{ and } L \ \models\ \varphi \quad \implies \quad (X, \varrho) \ \models\ \varphi$$

Prob As $L \ \models\ \varphi$ we know that $(X, \varrho) \models_L \varphi$. That is for any $\omega \in \text{ass}(X, \varrho)$ we have: If $(X, \varrho) \models_\omega \lambda$ for any $\lambda \in L$ then $(X, \varrho) \models_\omega \varphi$. But we also have $(X, \varrho) \ \models\ L$, that means $(X, \varrho) \models_\omega \lambda$ is satisfied for any $\omega \in \text{ass}(X, \varrho)$. Together we can conclude $(X, \varrho) \models_\omega \varphi$. But as this is true for any $\omega \in \text{ass}(X, \varrho)$ we have shown $(X, \varrho) \ \models\ \varphi$.

Proposition 36:
Let $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a (many-sorted, first-order) formal language and fix $(X, \varrho) \in \text{real}(\mathcal{L})$ a realization of $\mathcal{L}$. Then we get the property

(i) Also consider any two formulae φ, $\psi \in \text{form}(\mathcal{L})$, then the following two statements are true:

- If $(X, \varrho) \models \neg\varphi$ then we do *not* have $(X, \varrho) \models \varphi$. In this sense the models relation always is consistent.
- If we have $(X, \varrho) \models \varphi \to \psi$ and $(X, \varrho) \models \varphi$, then we also get $(X, \varrho) \models \psi$. In this sense we can pull out $\to$ as an implication.

Now let $1 \leq n \in \mathbb{N}$ and consider the formulae $\varphi_1, \dots, \varphi_n \in \text{form}(\mathcal{L})$ over $\mathcal{L}$. Finally let $J : \mathbb{B}^n \to \mathbb{B}$ be an n-ary junctor and identify the logical values and numbers $0 = $ *false* and $1 = $ *true*. Then also

(ii) Fix any assignment $\omega \in \text{ass}(X, \varrho)$, then the following two statements are equivalent:

(a) $(X, \varrho) \models_\omega J(\varphi_1, \dots, \varphi_n)$
(b) $J\big((X, \varrho) \models_\omega \varphi_1, \dots, (X, \varrho) \models_\omega \varphi_n\big)$

(iii) If the φ_k are sentences, i.e. $\text{free}(\varphi_k) = \emptyset$ for any $k \in 1 \dots n$, then even the following two statements are equivalent:

(a) $(X, \varrho) \models J(\varphi_1, \dots, \varphi_n)$
(b) $J\big((X, \varrho) \models \varphi_1, \dots, (X, \varrho) \models \varphi_n\big)$

Proof:

(i) If $(X, \varrho) \models \neg\varphi$ then by definition of $\models$ we have $(\neg\varphi)(\omega) = 1$ for any assignment $\omega \in \text{ass}(X, \varrho)$. But by the recursive definition of the evaluation this is $1 - \varphi(\omega) = 1$, which again is $\varphi(\omega) = 0 \neq 1$. As the set of assignments clearly is non-empty, this means, that $(X, \varrho) \models \varphi$ is not the case. Likewise, if $(X, \varrho) \models \varphi \to \psi$ then for any assignment $\omega \in \text{ass}(X, \varrho)$ we have

$$\max\{\, 1 - \varphi(\omega),\ \psi(\omega)\,\} \;=\; 1$$

So this is $\varphi(\omega) = 0$ or $\psi(\omega) = 1$. But by assumption we also have $\varphi(\omega) = 1$, so this only leaves $\psi(\omega) = 1$. And as this is true for any assignment ω we have $(X, \varrho) \models \psi$, as claimed.

(ii) In order to bring J into $\mathcal{L}$ we have to write it as a combination of $\neg$ and $\to$. So by induction on the number of construction steps to build up J it suffices to prove the equivalence for $\neg$ and $\to$ only. We start with $\neg$, proving

$$(X, \varrho) \models_\omega \neg\varphi \quad\Longleftrightarrow\quad \textit{not } (X, \varrho) \models_\omega \varphi$$

But this is evident $1 - \varphi(\omega) = 1$ is equivalent to $\varphi(\omega) = 0$ and hence to *not* $\varphi(\omega) = 1$, as $\mathbb{B}$ only contains these two numbers 0 and 1. So we need to check the following equivalence for $\to$ next

$$(X, \varrho) \models_\omega \varphi \to \psi \quad\Longleftrightarrow\quad \textit{if } (X, \varrho) \models_\omega \varphi \textit{ then } (X, \varrho) \models_\omega \psi$$

By definition $(X, \varrho) \models_\omega \varphi \to \psi$ is $\varphi(\omega) = 0$ or $\psi(\omega) = 1$. So there is exactly one case in which this is untrue: $\varphi(\omega) = 1$ and $\psi(\omega) = 0$. And this is precisely the case in which the statement *if* $\varphi(\omega) = 1$ *then* $\psi(\omega) = 1$ is untrue.

(iii) If φ does not have any free variables, then the truth of φ does not depend on the assignment ω. That is to say that $\forall\, \omega \in \mathrm{ass}(X, \varrho)$ we have $\varphi(\omega) = 1$ and $\exists\, \omega \in \mathrm{ass}(X, \varrho)$ such that $\varphi(\omega) = 1$ are equivalent, according to lemma 30. As $\mathrm{ass}(X, \varrho) \neq \emptyset$ (iii) follows from (ii) immediately.

$\square$

7 The Theory of Proofs

So far we have introduced two notions of proofs in a formal language $\mathcal{L}$, namely *deduction*, which we denote by $L \vdash \varphi$ and *implication*, which we mark by $L \models \varphi$. The aim of this section is to prove that in the end both notions are equivalent! That is if φ can be deduced from L then φ is also implied by L and vice versa. This is the deep reason behind the fact that mathematicians use both methods interchangably. The easy part is the Correctness Theorem, that guarantees that any deduction also gives rise to an implication. It is done again by induction on the length of the deduction. The hard part is the Completeness Theorem, that says that there also is a deduction to any implication. The proof of this is far from being constructive, in fact we use a zornification. In result we have absolutely no control over how long the deduction will be, nor how difficult it is to find. All we can tell is that *there is* a deduction - at least if we accept the axiom of choice. And we need a whole bunch of preparations in order to be able to conduct the proof. These preparations are called Henkin Theory which will be established in the next section 8.

Definition 37:
Let $\mathcal{L}$ be a (many-sorted, first order) formal language and recall that we denoted the set of all formulae of $\mathcal{L}$, that do not have free variables by $\mathrm{sen}(\mathcal{L}) = \{ \varphi \in \mathrm{form}(\mathcal{L}) \mid \mathrm{free}(\varphi) = \emptyset \}$. Then we further define

(i) A nonempty set $T \subseteq \mathrm{sen}(\mathcal{L})$ of sentences of $\mathcal{L}$ is called a **theory** in $\mathcal{L}$. A set $A \subseteq \mathrm{sen}(\mathcal{L})$ is said to be a set of **axioms** of the theory T, iff it satisfies $\mathrm{con}(A) = \mathrm{con}(T)$. And T is called **finitely axiomatizable**, iff there is a finite set of axioms for T.

(ii) For any theory $T \subseteq \mathrm{sen}(\mathcal{L})$ we define the **(deductive) closure** of T to be $\overline{T} := \mathrm{con}(T) \cap \mathrm{sen}(\mathcal{L})$. Now T is said to be a **(deductively) closed** theory, iff any sentence $\varphi \in \mathrm{sen}(\mathcal{L})$ with $T \vdash \varphi$ already is contained $\varphi \in T$. In other words, iff

$$T \;=\; \overline{T} \;:=\; \mathrm{con}(T) \cap \mathrm{sen}(\mathcal{L})$$

(iii) A theory $T \subseteq \mathrm{sen}(\mathcal{L})$ is **complete**, iff it satisfies one of the following three equivalent conditions:

(a) For any sentence $\varphi \in \mathrm{sen}(\mathcal{L})$ we can either deduce φ or its negation $\neg\varphi$ (but not both) from T, formally that is

$$\forall \varphi \in \mathrm{sen}(\mathcal{L}) \;:\; T \vdash \varphi \;\text{ xor }\; T \vdash \neg\varphi$$

(b) For any two sentences φ, $\psi \in \mathrm{sen}(\mathcal{L})$ we have the properties: (1) We have $L \vdash \neg\varphi$ if and only if *not* $L \vdash \varphi$ and (2) We have $L \vdash \varphi \to \psi$ if and only if $L \vdash \varphi$ implies $L \vdash \psi$.

(c) T is consistent but any theory $U \subseteq \mathrm{sen}(\mathcal{L})$ that allows to draw more conclusions than T already is contradictory, formally again

$$\forall U \subseteq \mathrm{sen}(\mathcal{L}) \;:\; \overline{T} \subset \overline{U} \implies U \text{ is contradictory}$$

(iv) If $\varphi \in \mathrm{form}(\mathcal{L})$ is a formula having precisely the free variables $\mathrm{free}(\varphi) = \{ x_i, \dots, x_n \}$ where x_k is of the sort $i_k := \mathrm{sort}(x_k)$, then we define the **universal closure** of φ to be the following sentence

$$\forall\!\!\!\forall\, \varphi \;:=\; \forall_{i_1} x_1 \dots \forall_{i_n} x_n\, \varphi \;\in\; \mathrm{sen}(\mathcal{L})$$

Proposition 38:
Let $\mathcal{L}$ be a (many-sorted, first order) formal language, $L \subseteq \text{form}(\mathcal{L})$ be a set of formulae and $T \subseteq \text{sen}(\mathcal{L})$ be a theory in $\mathcal{L}$, then we obtain

(i) For any formula $\varphi \in \text{form}(\mathcal{L})$ the universal closure of φ is logically equivalent to the original formula, that is
$$\varphi \quad \dashv\vdash \quad \forall\!\!\!\forall\, \varphi$$

(ii) If (X, ϱ) is a realization and $\varphi \in \text{form}(\mathcal{L})$ is a formula in $\mathcal{L}$, then we find the equivalence of
$$(X, \varrho) \models \varphi \iff (X, \varrho) \models \forall\!\!\!\forall\, \varphi$$

(iii) $\overline{L} = \text{con}(L) \cap \text{sen}(\mathcal{L})$ is a closed theory. If L is consistent, then so is $\overline{L}$.

(iv) For any formula $\varphi \in \text{form}(\mathcal{L})$, equivalent are: $L \vdash \varphi \iff \overline{L} \vdash \varphi$.

(v) If L and $M \subseteq \text{form}(\mathcal{L})$ are any sets of formulae of $\mathcal{L}$, then we say that $M : L$ is a **conservative extension** iff $L \subseteq M$ and we have one of the following two equivalent properties

   (a) $\overline{L} = \overline{M}$

   (b) $\text{con}(L) = \text{con}(M)$

Proof of 37:
We need to prove the equivalence of (a) to (c) in the definition of complete theories. Here (a) $\iff$ (b) can be taken literally from the proof of 28, we only restrict the argument to sentences here. For (a) $\implies$ (c) we first note that T is consistent: Choose any sort $i \in I$ and $x := x_{i,1} \in \text{var}(\mathcal{L})$ and $\varphi := \forall_i x(x =_i x)$, then we have $\emptyset \vdash \varphi$ by (L3) and (D5) and hence $T \vdash \varphi$ by (D2). Now by assumption (a) this means that we *do not* have $T \vdash \neg\varphi$ and hence property (d) of a contradictory set is ruled out.
Next we consider some $T \subseteq U \subseteq \text{sen}(\mathcal{L})$ such that $\overline{T} \subset \overline{U}$. That is there is some $\varphi \in \overline{U}$ with $\varphi \notin \overline{T}$. As $\varphi \in \text{sen}(\mathcal{L})$ this means $\varphi \in \text{con}(U)$ but $\varphi \notin \text{con}(T)$, in other words $U \vdash \varphi$ and *not* $T \vdash \varphi$. By assumption on T this means $T \vdash \neg\varphi$. But as $T \subseteq U$ this also implies $U \vdash \neg\varphi$ by (D2). Hence we have found some φ with both $U \vdash \varphi$ and $U \vdash \neg\varphi$ and hence U is contradictory due to property (c).
For (c) $\implies$ (a) we have to show that for any $\varphi \in \text{sen}(\mathcal{L})$ the deduction $T \vdash \varphi$ is equivalent to *not* $T \vdash \neg\varphi$. As T is consistent $T \vdash \varphi$ already implies *not* $T \vdash \neg\varphi$, as else T would satisfy property (c) of contradictory sets. It remains to show that *not* $T \vdash \varphi$ implies $T \vdash \neg\varphi$: So let us define $U := T \cup \{\neg\varphi\}$, by remark 27, this set is consistent, too. Hence by assumption we must have $\overline{U} = \overline{T}$, thus $\neg\varphi \in U \subseteq \overline{U} = \overline{T}$ yields $T \vdash \neg\varphi$. □

Proof of 38:

(i) By induction on n it suffices to check the case $\varphi \dashv\vdash \forall_i\, x\, \varphi$ for any formula with $x \in \text{free}(\varphi)$. To do this by 22 we have to prove both the deductions (a) $\emptyset \vdash \forall_i\, x\, \varphi \to \varphi$ and (b) $\emptyset \vdash \varphi \vdash \forall_i\, x\, \varphi$.

It is easy to deal with (a): The term $t := x$ contains the free variable x only. So in $\varphi' = Qx\, \varphi''$ the variable x can never be free. Hence x always is freely substitutable by $t = x$ and $\varphi[x : x] = \varphi$ is trivial. So (L6) allows us to write down $\emptyset \vdash \forall_i\, x\, \varphi \to \varphi[x : x]$ and this is (a).

Statement (b) is not much harder: By 17.(ii) we have $\emptyset \vdash \varphi \to \varphi$. And by the generalization rule (D5) this enables $\emptyset \vdash \forall_i\, x(\varphi \to \varphi)$. But we also have (L4) which allows $\emptyset \vdash \forall_i\, x(\varphi \to \varphi) \to (\varphi \to \forall_i\, x\,\varphi)$. Together with modus ponens (D3) this is (b).

(ii) We will prove the statement by induction on the number n of free variables of φ. In case $n = 0$ we have $\forall\!\!\!\forall\, \varphi = \varphi$ and hence there is nothing to prove. For the induction step it suffices to prove

$$(X, \varrho) \models \varphi \iff (X, \varrho) \models \forall_i\, x\,\varphi$$

where x is any variable symbol of the sort$(x) =: i$. But this is rather easy: By definition $(X, \varrho) \models \forall_i\, x\,\varphi$ if and only if for any $\omega \in \mathrm{ass}(X, \varrho)$ we get $(\forall_i\, x\,\varphi)(\omega) = 1$. But by definition this again is equivalent, to $\{\, a \in X_i \mid \varphi(\omega[x : a]) = 1 \,\} = X_i$. Naive set theory now tells us this is $\forall\, a \in X_i : \varphi(\omega[x : a]) = 1$. But any $a \in X_i$ means all assignments ω and hence this is equivalent, to $\forall\, \omega \in \mathrm{ass}(X, \varrho) : \varphi(\omega) = 1$ which again is nothing but $(X, \varrho) \models \varphi$.

(iv) We start with the implication $\overline{L} \vdash \varphi \implies L \vdash \varphi$: So consider any formula $\varphi \in \mathrm{form}(\mathcal{L})$ with $\overline{L} \vdash \varphi$. As $\overline{L} \subseteq \mathrm{con}(L)$ by construction we also have $\mathrm{con}(\overline{L}) \subseteq \mathrm{con}(\mathrm{con}(L))$ by the dilution rule (D2). But we have already seen in remark 27 that $\mathrm{con}(\mathrm{con}(L)) = \mathrm{con}(L)$, so altogether we have $\varphi \in \mathrm{con}(\overline{L}) \subseteq \mathrm{con}(L)$ and hence $L \vdash \varphi$.

For the converse implication $L \vdash \varphi \implies \overline{L} \vdash \varphi$ we note that $L \vdash \varphi$ also implies $L \vdash \forall\!\!\!\forall\, \varphi$ according to (i). That is $\forall\!\!\!\forall\, \varphi \in \mathrm{con}(L)$ and as it also is a sentence $\forall\!\!\!\forall\, \varphi \in \mathrm{sen}(\mathcal{L})$ we find $\forall\!\!\!\forall\, \varphi \in \mathrm{con}(L) \cap \mathrm{sen}(\mathcal{L}) = \overline{L}$. By the trivial rule (D1) this of course yields $\overline{L} \vdash \forall\!\!\!\forall\, \varphi$. And using (i) again we thereby arrive at $\overline{L} \vdash \varphi$.

(iii) First of all $\overline{L} \subseteq \mathrm{sen}(\mathcal{L})$ is a theory by construction. We need to show that it also is a closed theory. So consider any sentence $\varphi \in \mathrm{sen}(\mathcal{L})$ with $\overline{L} \vdash \varphi$, then we need to show $\varphi \in \overline{L}$. By (iii) $\overline{L} \vdash \varphi$ implies $L \vdash \varphi$ and hence we have $\varphi \in \mathrm{con}(L)$. As we started with a sentence $\varphi \in \mathrm{sen}(\mathcal{L})$ this now means $\varphi \in \mathrm{con}(L) \cap \mathrm{sen}(\mathcal{L}) = \overline{L}$.

Now if L is consistent, then we need to show that $\overline{L}$ is consistent, too. Suppose $\overline{L}$ was contradictory, then we could deduce $\overline{L} \vdash \neg(x =_i x)$ for some (in fact any) variable $x \in \mathrm{var}(\mathcal{L})$. By (iv) again this would imply $L \vdash \neg(x =_i x)$. In turn this would mean that L was contradictory from the start.

(v) If $\mathrm{con}(L) = \mathrm{con}(M)$ then $\overline{L} = \overline{M}$ is trivial. Conversely suppose $\overline{L} = \overline{M}$ and consider any formula $\varphi \in \mathrm{form}(\mathcal{L})$ such that $L \vdash \varphi$. Then by (i) we also have $L \vdash \forall\!\!\!\forall\, \varphi$ and hence $\forall\!\!\!\forall\, \varphi \in \overline{L} = \overline{M}$. Now $\forall\!\!\!\forall\, \varphi \in \overline{M}$ translates into $M \vdash \forall\!\!\!\forall\, \varphi$ and by (i) again, this is $M \vdash \varphi$. That is we have proved $\mathrm{con}(L) \subseteq \mathrm{con}(M)$ and by the symmetry of the argument $\mathrm{con}(L) = \mathrm{con}(M)$.

□

Theorem 39:
[AC] Let $T \subseteq \text{sen}(\mathcal{L})$ be a theory and $L \subseteq \text{form}(\mathcal{L})$ be any set of formulae of the formal language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$, then we find the statements

(i) T is consistent if and only if it is realizable, that is to say the following two statements are equivalent

(a) T is **consistent**, i.e. $\nexists \varphi \in \text{form}(\mathcal{L})$ with $T \vdash \varphi$ and $T \vdash \neg\varphi$

(b) T is **realizable**, i.e. $\exists\, (X, \varrho)$ $\mathcal{L}$-structure, with $(X, \varrho) \models T$

(ii) Let T be a closed, consistent theory. Then T is complete, iff any two models of T are *elementary equivalent*. Formally that is the equivalence of the following two statements

(a) $\forall\, \varphi \in \text{sen}(\mathcal{L})$ we get $\varphi \in T$ xor $\neg\varphi \in T$

(b) $\forall\, (X, \varrho), (Y, \sigma) \in \text{model}(T)$ we get $\text{con}(X, \varrho) = \text{con}(Y, \sigma)$

(iii) **Theorem of Lindenbaum**
If T is consistent, then there is a closed, complete theory $\widetilde{T} \subseteq \text{sen}(\mathcal{L})$ containing T, that is $T \subseteq \widetilde{T}$.

(iv) **Correctness and Completeness Theorem**
Any formula $\varphi \in \text{form}(\mathcal{L})$ can be deduced from L if and only if L implies φ. That is we have the following equivalence

$$L \vdash \varphi \quad \Longleftrightarrow \quad L \models \varphi$$

(v) **Compactness Theorem**
For any formula $\varphi \in \text{form}(\mathcal{L})$ we find that $L \models \varphi$ if and only if there is a finite subset $L_0 \subseteq L$ such that $L_0 \models \varphi$. And hence L is realizable if and only if any finite subset $L_0 \subseteq L$ is realizable.

(vi) If $(X, \varrho) \in \text{real}(\mathcal{L})$ is any realization of $\mathcal{L}$, then $\text{con}(X, \varrho) \cap \text{sen}(\mathcal{L})$ is a closed, consistent and complete theory of $\mathcal{L}$. And if $(X, \varrho) \in \text{model}(L)$ is a model of L then also find the inclusions

$$L \;\subseteq\; \text{con}(L) \;\subseteq\; \text{con}(X, \varrho)$$

Proof:

(iii) This proof will be an easy zornification, to do this we start with the set $\mathcal{Z}$ of all consistent theories T' that contain T, formally

$$\mathcal{Z} \;:=\; \left\{\, T' \mid T \subseteq T' \subseteq \text{sen}(\mathcal{L}) \text{ and } T' \text{ is consistent} \,\right\}$$

Obviously $\mathcal{Z}$ is partially ordered under the inclusion "$\subseteq$" and it is nonempty, since $T \in \mathcal{Z}$. Let now T_λ (where $\lambda \in \Lambda$) be some chain in $\mathcal{Z}$, then we define

$$T_\Lambda \;:=\; \bigcup_{\lambda \in \Lambda} T_\lambda$$

Clearly T_Λ is a theory of $\mathcal{L}$, but it is consistent, too - suppose there was some $\gamma \in \mathrm{form}(\mathcal{L})$ with both $T_\Lambda \ \vdash\ \gamma \wedge (\neg\gamma)$. Since a deduction has finitely many steps only, there are finitely many indexes $\lambda_1, \dots, \lambda_n \in \Lambda$ such that we already had

$$T_{\lambda_1} \cup \dots \cup T_{\lambda_n} \ \vdash\ \gamma \wedge (\neg\gamma)$$

Now choose $\mu \in \{ \lambda_1, \dots, \lambda_n \} \subseteq \Lambda$ in such a way that T_μ is maximal among $T_{\lambda_1}, \dots, T_{\lambda_n}$ (which is possible, as a chain is totally ordered). Then $T_\mu \ \vdash\ \gamma \wedge (\neg\gamma)$ would already be inconsistent in contradiction to $T_\mu \in \mathcal{Z}$. Thus there can be no such γ which means that $T_\Lambda \in \mathcal{Z}$ and hence - by Zorn's Lemma - we find a maximal element $T^* \in \mathcal{Z}$. Now define $\widetilde{T}$ to be the closure of some maximal element T^* of $\mathcal{Z}$

$$\widetilde{T} \ :=\ \overline{T^*} \ =\ \mathrm{con}(T^*) \cap \mathrm{sen}(\mathcal{L})$$

Then $\widetilde{T}$ is a closed theory, according to 38.(iii). And as $T^* \in \mathcal{Z}$ we find that T^* is consistent such that 38.(iii) also tells us, that $\widetilde{T}$ is consistent, too. We will now prove, that $\widetilde{T}$ even is complete. To do this suppose we have $\widetilde{T} \ \not\vdash\ \varphi$ for some sentence $\varphi \in \mathrm{sen}(\mathcal{L})$, then we take to

$$\widehat{T} \ :=\ \widetilde{T} \cup \{ \varphi \}$$

By assumption we have $T^* \subseteq \widetilde{T} \subset \widehat{T}$, but since T^* is maximal among the consistent theories, this implies, that $\widehat{T}$ is contradictory. In particular $\widetilde{T} \cup \{ \varphi \} = \widehat{T} \ \vdash\ \neg\varphi$. But since also $\widetilde{T} \cup \{ \neg\varphi \} \ \vdash\ \neg\varphi$ the section rule yields $\widetilde{T} \ \vdash\ \neg\varphi$. That is we have

$$\widetilde{T} \not\vdash \varphi \quad \Longrightarrow \quad \widetilde{T} \vdash \neg\varphi$$

But as $\widetilde{T}$ is consistent, we can never have both $\widetilde{T} \ \vdash\ \varphi$ and $\widetilde{T} \ \vdash\ \neg\varphi$. Therefore we either have one $\widetilde{T} \ \vdash\ \varphi$ or the other $\widetilde{T} \ \vdash\ \neg\varphi$, but not both, which means that $\widetilde{T}$ also is a complete theory.

(iv) We will first proof the *Correctness Theorem*, i.e. the direction that $L \ \vdash\ \varphi$ also implies $L \ \models\ \varphi$. That is we start with any realization (X, ϱ) and any assignment $\omega \in \mathrm{ass}(X, \varrho)$ then we need to show, that if for any $\lambda \in L$ we have $\lambda(\omega) = 1$ then we also get $\varphi(\omega) = 1$. This is done by induction on the length of the deduction $L \ \vdash\ \varphi$. To found the induction we need to regard the logical axioms and the trivial rule. For the induction step we need to imitate (D2) to (D5) for $\models$:

- Let us give a short preparation: Recall that we write $\varphi(\omega) \in \mathbb{B}$ and identify $\mathbb{B} = \mathbb{Z}_2$ for the interpretation of the formula φ under the assignment ω. And by definition we have

$$\begin{aligned} (\varphi \to \psi)(\omega) &= \max\{ 1 - \varphi(\omega), \psi(\omega) \} \\ &= \begin{cases} 1 & \text{if } \varphi(\omega) = 1 \text{ and } \varphi(\omega) = 0 \\ 0 & \text{in any other case} \end{cases} \end{aligned}$$

  Thus $(\varphi \to \psi)(\omega)$ equals 1 if and only if the implication $\varphi(\omega) = 1 \implies \psi(\omega) = 1$ of naive logic is true. Next we will prove that $(\varphi \leftrightarrow \psi)(\omega)$ equals 1 if and only if the equivalence of naive logic $\varphi(\omega) = 1 \iff \psi(\omega) = 1$ is true, formally

$$(\varphi \leftrightarrow \psi)(\omega) \;=\; \begin{cases} 1 \text{ if } \varphi(\omega) = \psi(\omega) \\ 0 \text{ if } \varphi(\omega) \neq \psi(\omega) \end{cases}$$

To do this we need to recall the definition of $\varphi \leftrightarrow \psi$ in the language $\mathcal{L}$ as $\neg(\gamma \to \neg\delta)$ where $\gamma := \varphi \to \psi$ and $\delta := \psi \to \varphi$. If we abbreviate $A := \varphi(\omega)$ and $B := \psi(\omega) \in \mathbb{Z}_2$ then $(\varphi \to \psi)(\omega) = 1 - A(1-B)$ and thereby we compute

$$\begin{aligned} \big(\neg(\gamma \to \neg\delta)\big)(\omega) &= 1 - \big(1 - \gamma(\omega)\delta(\omega)\big) \;=\; \gamma(\omega)\delta(\omega) \\ &= \big(1 - A(1-B)\big)\big(1 - B(1-A)\big) \\ &= 1 - A - B + 3AB - AB^2 - A^2B + A^2B^2 \end{aligned}$$

In $\mathbb{Z}_2$ we have $A^2 = A$ and $2A = 0$, so this reduces to $1 - A - B$ and this expression equals 1 iff $(A,B) = (0,0)$ or $(A,B) = (1,1)$ and equals 0 for $(A,B) = (0,1)$ or $(A,B) = (1,0)$.

- For **(L1)** we start with $\emptyset \vdash \varphi \to (\psi \to \varphi)$ and let $A := \varphi(\omega)$ and $B := \psi(\omega) \in \mathbb{Z}_2$. Then we compute

$$\begin{aligned} \Big(\varphi \to (\psi \to \varphi)\Big)(\omega) &= 1 - \varphi(\omega)\big(1 - (\psi \to \varphi)(\omega)\big) \\ &= 1 - A\big(1 - (1 - B(1-A))\big) \\ &= 1 - AB + A^2B \end{aligned}$$

  But as we are in $\mathbb{Z}_2$ we have $A^2 = A$ and hence the evaluation of $\varphi \to (\psi \to \varphi)$ is one for any $\omega \in \text{ass}(X,\varrho)$. The same is true for **(L2)** which reads as $\emptyset \vdash (\neg\varphi) \to (\varphi \to \psi)$

$$\begin{aligned} \Big((\neg\varphi) \to (\varphi \to \psi)\Big)(\omega) &= 1 - \big(1 - \varphi(\omega)\big)\big(1 - (\varphi \to \psi)(\omega)\big) \\ &= 1 - \big(1 - A\big)\big(1 - (1 - A(1-B))\big) \\ &= 1 - A + A^2 + AB - A^2B \end{aligned}$$

  Again $-A$ and $A^2 = A$ cancel each other and so do AB and $-A^2B$, such that we end up with a universally true formula, again. For **(L5)** we need to check (cf. the preparation above)

$$(\forall_i\, x\, \varphi)(\omega) \;=\; (\neg\exists_i\, x\, \neg\varphi)(\omega)$$

  Let us abbreviate $\varphi(a) := \varphi(\omega[x:a])$ then the left hand side is 1 iff $X_i = \{\, a \in X_i \mid \varphi(a) = 1 \,\}$ and the right hand side is 1 iff $\emptyset = \{\, a \in X_i \mid \varphi(a) = 0 \,\}$. As $\varphi(a)$ can only be either 1 or 0 these properties are truly equivalent.

- Rule **(L3)** is trivial, as $\omega(x) = \omega(x)$ for any variable symbol $x \in \text{var}(\mathcal{L})$ the formula $(x =_i x)$ is evaluated to 1. Likewise **(L7)** is evaluated to $1 - (\omega(x) = \omega(y)) \cdot (1 - (\varphi(\omega) = \varphi'(\omega)))$. In case $\omega(x) \neq \omega(y)$ there only remains 1 and in case $\omega(x) = \omega(y)$ we have $\varphi(\omega) = \varphi'(\omega)$, as in φ' we only replaced some x for y. So in this case only 1 remains, as well.

- For **(L4)** we need to show that the following interpretation equals 1. To do this we use the property $\max\{A, \max\{B, C\}\} = \max\{A, B, C\}$ of the maximum and commence:

$$\begin{aligned}
&\Big(\forall_i\, x\, (\varphi \to \psi) \to (\varphi \to \forall_i\, x\, \psi)\big)\Big)(\omega) \\
=\;& \max\big\{ 1 - \big(\forall_i\, x\, (\varphi \to \psi)\big)(\omega), \big(\varphi \to \forall_i\, x\, \psi\big)(\omega) \big\} \\
=\;& \max\big\{ 1 - \big(\forall_i\, x\, (\varphi \to \psi)\big)(\omega), 1 - \varphi(\omega), \big(\forall_i\, x\, \psi\big)(\omega) \big\}
\end{aligned}$$

  If $(\forall_i\, x\, (\varphi \to \psi))(\omega) = 0$ or $\varphi(\omega) = 0$ then the whole expression will be 1 so there only remains the case where both are 1. By definition of the interpretation this simply stands for

$$\{ a \in X_i \mid \varphi(a) \implies \psi(a) \} = X_i = \{ a \in X_i \mid \varphi(a) \}$$

  As before $\psi(a)$ abbreviates $\psi(\omega[x : a])$). And together these yield $X_i = \{ a \in X_i \mid \psi(a) \}$, in other words $(\forall_i\, x\, \psi)(\omega) = 1$.

- For **(L6)** we choose the same approach, as for (L4), only that we have to regard the following interpretation of

$$\big(\forall_i\, x\, \varphi \to \varphi[x : t]\big)(\omega) \;=\; \max\big\{ 1 - \big(\forall_i\, x\, \varphi\big)(\omega), \big(\varphi[x : t]\big)(\omega) \big\}$$

  Again we only need to consider $(\forall_i\, x\, \varphi)(\omega) = 1$, i.e. we have $X_i = \{ a \in X_i \mid \varphi(a) \}$. Let now $b := t(\omega) \in X_i$, then this yields $\varphi(b) = 1$. But as x is freely substitutable by t we can use proposition 31 which gives $\varphi[x : t](\omega) = \varphi(\omega[x : b]) = \varphi(b) = 1$.

- By now we have shown all the logical axioms. It remains to verify the deduction rules. We start by **(D1)** which really is trivial - for any $\varphi \in L$ we already have $\varphi(\omega) = 1$ by assumption.

- In the case of **(D2)** - by the induction hypothesis - we already know $L \models \varphi$. But by definition of the $\models$-relation, $L \cup M \models \varphi$ then is trivial.

- In case of the modus ponens **(D3)** we - by induction hypothesis - have $L \models \varphi$ and $L \models \varphi \to \psi$. By our preparation above

$$\big(\varphi \to \psi\big)(\omega) = 1 \iff \Big(\varphi(\omega) = 1 \implies \psi(\omega) = 1\Big)$$

  So for any ω with $\forall\, \lambda \in L : \lambda(\omega) = 1$ we have both $\varphi(\omega)$ and $\varphi(\omega) \implies \psi(\omega)$. Thus applying the modus ponens in naive logic yields $\psi(\omega) = 1$ and hence $L \models \psi$.

- Regarding the section rule **(D4)** the induction hypothesis guarantees $L \cup \{ \varphi \} \models \psi$ and $L \cup \{ \neg\varphi \} \models \psi$. For a fixed realization (X, ϱ) and assignment ω, together these yield

$$\left.\begin{array}{l} \big(\forall\, \lambda \in L : \lambda(\omega) = 1 \text{ and } \varphi(\omega) = 1\big) \text{ and} \\ \big(\forall\, \lambda \in L : \lambda(\omega) = 1 \text{ and } \varphi(\omega) = 0\big) \end{array}\right\} \implies \psi(\omega) = 1$$

  As $\varphi(\omega)$ either is 0 or 1 we see that the value of $\varphi(\omega)$ doesn't really matter. Hence we conclude $\psi(\omega) = 1$ from the assumption $\forall\, \lambda \in L : \lambda(\omega) = 1$ alone and this is $L \models \psi$. As this is true for any $\lambda \in L$ the induction hypothesis $L \models \varphi$ implies $\varphi(\varepsilon)$ for any assignment ν. Taking $\nu := \omega[x : a]$ this is $\varphi(\omega[x : a]) = 1$.

- For **(D5)** the induction hypothesis conjures up $L \models \varphi$ and we need to prove $L \models \forall_i\, x\, \varphi$. Hereby x is a variable symbol that (case1) is $x \notin \text{free}(\varphi)$ or (case2) $\forall\, \lambda \in L : x \notin \text{free}(\lambda)$. So let us take a closer look at our *assumption* $L \models \varphi$, explicitly

$$\begin{cases} \forall\, (X, \varrho) \in \text{real}(\mathcal{L})\ \forall\, \omega \in \text{ass}(X, \varrho) \\ \big(\forall\, \lambda \in L : \lambda(\omega) = 1\big) \implies \varphi(\omega) = 1 \end{cases}$$

What we need to show is $L \models \forall_i\, x\, \varphi$. Again let us take a closer look: As $(\forall_i\, x\, \varphi)(\omega) = 1$ iff $\{\, a \in X_i \mid \varphi(\omega[x : a]) = 1 \,\} = X_i$ it turns out that our aim is to prove

$$\begin{cases} \forall\, (X, \varrho) \in \text{real}(\mathcal{L})\ \forall\, \omega \in \text{ass}(X, \varrho) \\ \big(\forall\, \lambda \in L : \lambda(\omega) = 1\big) \implies \forall\, a \in X_i : \varphi(a) = 1 \end{cases}$$

where we let $\varphi(a) = \varphi(\omega[x : a])$ again. Let us also abbreviate $\eta := \omega[x : a]$ that is $\varphi(a) = \varphi(\eta)$. In (case1) $\omega(y) = \eta(y)$ for any $y \in \text{free}(\varphi)$ and hence proposition 30 tells us $\varphi(a) = \varphi(\eta) = \varphi(\omega) = 1$. As this is true for any $a \in X_i$ we are done. In (case2) we have $\eta(y) = \omega(y)$ for any $y \in \text{free}(\lambda)$ where $\lambda \in L$ is arbitrary and likewise we get $\forall\, \lambda \in L : \lambda(\eta) = \lambda(\omega) = 1$. Using the assumption on η instead of ω we get $\varphi(a) = \varphi(\eta) = 1$. Again a could be any element of X_i, so we are done in this case, too.

(i) Using the Correctness Theorem the proof of (b) $\implies$ (a) is straightforward: Suppose (X, ϱ) is a model of T, but T was inconsistent, i.e. $T \vdash \gamma \wedge \neg\gamma$ for some formula $\gamma \in \text{form}(\mathcal{L})$. Then by (iv) above we get $T \models \gamma \wedge \neg\gamma$, but since $(X, \varrho) \models T$ this implies $(X, \varrho) \models \gamma \wedge \neg\gamma$. By definition of the interpretation, this would imply (for any assignment $\omega \in \text{ass}(X, \varrho)$)

$$1 \;=\; \text{val}^{\omega}_{(X,\varrho)}(\gamma \wedge \neg\gamma) \;=\; \text{val}^{\omega}_{(X,\varrho)}(\gamma) \cdot \big(1 - \text{val}^{\omega}_{(X,\varrho)}(\gamma)\big) \;=\; 0$$

Hence our assumption - the inconsistency of T - is false and the implication (b) $\implies$ (a) is established. The proof of (a) $\implies$ (b) requires Henkin's Lemma that explicitly constructs models for certain theories. We will present this lemma in the next section only, so the reader might want to postpone reading the proof until she is familiar with this lemma.
Since T is consistent, $\text{hen}(T)$ is a closed, consistent, henkinian theory according to 44.(iv). Denote by $T^* := (\text{hen}(T))^\sim$ the completion of $\text{hen}(T)$, which exists, by (iii). Since $\text{hen}(T) \subseteq T^*$ 44.(i) tells us that T^* not only is complete, but henkinian again. Thus by 45 T^* has a model (X^*, ϱ^*) in $\text{hen}(\mathcal{L}) : \mathcal{L}$. And as we have $T \subseteq T^* \cap \text{sen}(\mathcal{L})$ it is clear that the restriction of (X^*, ϱ^*) to $\mathcal{L}$ is a model of T

$$(X, \varrho) \;:=\; \Big(X^*,\, \varrho^*\big|_{\mathcal{L}}\Big)$$

(iv) Now we are able to prove the Completeness Theorem, i.e. the direction $\Longleftarrow$ of the equivalence. In this first step we will regard the case of a theory $T \subseteq \text{sen}(\mathcal{L})$ - the general case $L \subseteq \text{form}(\mathcal{L})$ we be treated in the subsequent step. Suppose $T \models \varphi$, then we will prove $T \vdash \varphi$ by contradiction, that is we assume $T \nvdash \varphi$ and continue

$$\begin{aligned} T \nvdash \varphi \quad &\implies \quad T \nvdash \forall\!\!\!\forall\, \varphi, \text{ as } \varphi \dashv\vdash \forall\!\!\!\forall\, \varphi \\ &\implies \quad T \cup \{\, \neg \forall\!\!\!\forall\, \varphi \,\} \text{ is a consistent theory} \end{aligned}$$

$$
\begin{aligned}
&\Longrightarrow \quad T \cup \{\, \neg \forall\!\!\!\forall\, \varphi \,\} \text{ has a model } (X,\varrho), \text{ by (i)} \\
&\Longrightarrow \quad (X,\varrho) \models T \text{ and } (X,\varrho) \models \neg \forall\!\!\!\forall\, \varphi \\
&\Longrightarrow \quad (X,\varrho) \models \varphi, \text{ as } (X,\varrho) \models T \text{ and } T \models \varphi \\
&\Longrightarrow \quad (X,\varrho) \models \forall\!\!\!\forall\, \varphi, \text{ by 38.(ii)}
\end{aligned}
$$

But by 36.(i) $(X,\varrho) \models \neg \forall\!\!\!\forall\, \varphi$ implies *not* $(X,\varrho) \models \forall\!\!\!\forall\, \varphi$ so we have found a contradiction, that can only be solved by abandoning $T \nvdash \varphi$ and accepting $T \vdash \varphi$.

(iv) Now we will prove the Completeness Theorem for general $L \subseteq \text{form}(\mathcal{L})$. It stands to reason to reduce the general case to the case of theories. This can be done by introducing new constant symbols to replace all the free variable symbols in L and φ. Thus we start with $L \models \varphi$ and define the index sets $J_i \subseteq \mathbb{N}$ (for $i \in I$) by demanding

$$\{\, x_{i,j} \mid i \in I, j \in J_i \,\} \;=\; \text{free}(\varphi) \cup \bigcup_{\lambda \in L} \text{free}(\lambda)$$

Without loss of generality we may assume that the variable symbols are used in the following way: Free variables of $L \cup \{\, \varphi \,\}$ have indexes j that are divisible by 3, that is $J_i \subseteq 3\mathbb{N}$ and bound variables of $L \cup \{\, \varphi \,\}$ have indexes that are 1 modulo 3. Thereby bound variable symbols do not appear in the J_i. We now pick up new constant symbols

$$c_{i,j} \text{ of sort}^*(c_{i,j}) := i \text{ for any } i \in I, j \in J_i$$

where we demand that $c_{i,j} \notin \mathcal{C}$ are symbols, that have not been used before. Now the language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ is expanded by these

$$\mathcal{L}^* \;:= \mathcal{L}_I(\mathcal{C} \cup \{\, c_{i,j} \mid i \in I, j \in J_i \,\}, \mathcal{F}, \mathcal{R}, \text{sort}^*)$$

For any $\lambda \in L \cup \{\, \varphi \,\}$ let λ^* denote the formula, that is obtained by replacing any $x_{i,j}$ by the respective constant symbol $c_{i,j}$

$$\lambda^* \;:= \lambda\big(x_{i,j} : c_{i,j} \mid i \in I, j \in J_i\big)$$

Then λ^* obviously is a statement of $\mathcal{L}^*$. Let us denote the theory $L^* := \{\, \lambda^* \mid \lambda \in L \,\} \subseteq \text{sen}(\mathcal{L}^*)$, then we will prove the claim in two more steps:

Claim $L \models \varphi \quad \Longrightarrow \quad L^* \models \varphi^*$
Proof since L^* is a theory assignments are irrelevant and it suffices to check the following implication for all realizations $(X,\varrho)^*$ of $\mathcal{L}^*$

$$\forall\, \lambda \in L : (X,\varrho)^* \models \lambda^* \quad \Longrightarrow \quad (X,\varrho)^* \models \varphi^*$$

Let now ω^* be an assignment with $\omega^*(x_{i,j}) = \omega^*(c_{i,j})$ for $i \in I$ and $j \in J_i$. By assumption we have $\forall\, \lambda \in L : (X,\varrho)^* \models \lambda^*$ in particular $\forall\, \lambda \in L : (X,\varrho)^* \models_{\omega^*} \lambda^*$. By the assumption $L \models \varphi$ and the construction of ω^* this implies $(X,\varrho)^* \models_{\omega} \varphi$ and hence $(X,\varrho)^* \models \varphi^*$.

Now that we have $L^* \models \varphi^*$ we may apply the Completeness Theorem in the (above) case of theories to obtain $L^* \vdash \varphi^*$. The final step for the general theorem is to dedcuce $L \vdash \varphi$ from this which we will do next:

Claim $L^* \vdash \varphi^* \implies L \vdash \varphi$
Proof since the deduction of φ^* has finitely many steps only, it can be based on finitely many formulae $\lambda_1^*, \dots, \lambda_n^* \in L^*$. We may assume that none these formulae λ_k^* contains any of the variable symbols $x_{i,j}$ for $i \in I$ and $j \in J_i$, e.g. by renaming all their variable symbols $x_{i,j} \mapsto x_{i,3j+2}$ such that all the indexes j of these symbols are 2 modulo 3. It is clear that any deduction step in $\mathcal{L}^*$ is turned into the same deduction step in $\mathcal{L}$ by eliminating all the new variable symbols by letting $\lambda_k := \lambda_k^*[c_{i,j} : x_{i,j} \mid i \in I, j \in J_i]$. And hence we have found a deduction of φ in $\mathcal{L}$, namely

$$\{ \lambda_1, \dots, \lambda_n \} \vdash \varphi$$

Nota in the two claims above the converse implication is true, as well.

(ii) The proof of (a) $\implies$ (b) is straightforward: Let (X, ϱ) and (Y, σ) be any two models of T and $\varphi \in \mathrm{sen}(\mathcal{L})$ be any sentence. Then we need to show the equivalence

$$(X, \varrho) \models \varphi \iff (Y, \sigma) \models \varphi$$

In the case $T \vdash \varphi$ (and $T \not\vdash \varphi$) we get $T \models \varphi$ by (iv) and hence $(X, \varrho) \models \varphi$, as (X, ϱ) was assumed to be a model of T. Likewise $(Y, \sigma) \models \varphi$. If conversely $T \vdash \neg\varphi$ (and $T \not\vdash \varphi$) then again $(X, \varrho) \models \neg\varphi$ and $(Y, \sigma) \models \neg\varphi$. However, since φ is a sentence, by 36 this is equivalent, to $(X, \varrho) \not\models \varphi$ and $(Y, \sigma) \not\models \varphi$. So in any case we have the equivalence in (b).

In the converse direction (b) $\implies$ (a) we start with a closed, consistent theory T such that any two models of T are elementary equivalent. Now let $\varphi \in \mathrm{sen}(\mathcal{L})$, then we need to show that $\varphi \in T$ xor $\neg\varphi \in T$. First suppose $\varphi \in T$, then $\neg\varphi \notin T$, since T is consistent. Thus suppose $\varphi \notin T$, then (since T is closed and consistent) we find $T \not\vdash \varphi$ and hence $T \cup \{ \neg\varphi \}$ is consistent. By (i) $T \cup \{ \neg\varphi \}$ thereby has a model, say $(X, \varrho) \models T \cup \{ \neg\varphi \}$. This clearly is a model of T, too $(X, \varrho) \models T$ and it satisfies $(X, \varrho) \models \neg\varphi$. But by assumption (b) any model (Y, σ) of T has the same consequences, so for any model (Y, σ) of T we also find $(Y, \sigma) \models \neg\varphi$. Explicitly this reads

$$\forall (Y, \sigma) \in \mathrm{real}(\mathcal{L}) : \big((Y, \sigma) \models T \implies (Y, \sigma) \models \neg\varphi \big)$$

However we want to prove $T \models \neg\varphi$, that is for any realization (Y, σ) of $\mathcal{L}$ and any assignment $\omega \in \mathrm{ass}(Y, \sigma)$ we need to show

$$\big(\forall \tau \in T : (Y, \sigma) \models_\omega \tau \big) \implies (Y, \sigma) \models_\omega \neg\varphi$$

Thus suppose we are given a realization (Y,σ) and an assignment ω thereof, such that $(Y,\sigma) \models_\omega \tau$ for any $\tau \in T$. As T is a theory the specific assignment ω is irrelevant and hence we have $(Y,\sigma) \models_\omega \tau$ for any $\omega \in \mathrm{ass}(Y,\sigma)$ and any $\tau \in T$, but this is $(Y,\sigma) \models T$. And hence we get $(Y,\sigma) \models \neg\varphi$ by the implication above, now $(Y,\sigma) \models_\omega \neg\varphi$ is just a special case of this. Thus we have proved $T \models \neg\varphi$ and the Completeness Theorem (iv) grants $T \vdash \neg\varphi$. As T is a closed theory and φ is a sentence this finally is $\neg\varphi \in T$.

(v) The first statement is clear - for the deduction $L \vdash \varphi$: If $L_0 \subseteq L$ and $L_0 \vdash \varphi$, then $L \vdash \varphi$ is granted by the dilution rule (D2). Conversely if $L \vdash \varphi$ the deduction relation " $\vdash$ " only allows finitely many steps and hence only finitely many formulae $\lambda_1,\dots,\lambda_n$ on the left hand side are used. Now let $L_0 := \{\lambda_1,\dots,\lambda_n\} \subseteq L$ then we also have $L_0 \vdash \varphi$. The rest is an application of (iv)

$$
\begin{aligned}
L \models \varphi \quad &\iff \quad L \vdash \varphi \\
&\iff \quad \exists\, L_0 : L_0 \vdash \varphi \\
&\iff \quad \exists\, L_0 : L_0 \models \varphi
\end{aligned}
$$

The second claim is an ultimate consequence of the first combined with (i): Simply pursue the negation of the following equivalences

$$
\begin{aligned}
L \text{ not realizable} \quad &\iff \quad L \text{ contradictory} \\
&\iff \quad L \vdash \gamma \wedge \neg\gamma \\
&\iff \quad \exists\, L_0 : L_0 \vdash \gamma \wedge \neg\gamma \\
&\iff \quad \exists\, L_0 : L_0 \text{ contradictory} \\
&\iff \quad \exists\, L_0 : L_0 \text{ not realizable}
\end{aligned}
$$

(vi) Recall $\mathrm{con}(X,\varrho) = \{\varphi \in \mathrm{form}(\mathcal{L}) \mid (X,\varrho) \models \varphi\}$ and let us denote the theory $T := \mathrm{con}(X,\varrho) \cap \mathrm{sen}(\mathcal{L})$. By definition of $\mathrm{con}(X,\varrho)$ we have $(X,\varrho) \models \mathrm{con}(X,\varrho)$ and as $T \subseteq \mathrm{con}(X,\varrho)$ this yields $(X,\varrho) \models T$. Thus by (i) we know that T is a consistent theory. We will next prove that T also is a closed theory: Let $\varphi \in \mathrm{sen}(\mathcal{L})$ be a sentence with $T \vdash \varphi$. Then by (iv) we also have $T \models \varphi$ and as $(X,\varrho) \models T$ together this implies $(X,\varrho) \models \varphi$. But this is $\varphi \in \mathrm{con}(X,\varrho) \cap \mathrm{sen}(\mathcal{L}) = T$ again.

It remains to prove that T is a complete theory: Suppose we have $\varphi \notin T$ for some $\varphi \in \mathrm{sen}(\mathcal{L})$. By construction of T this means $\varphi \notin \mathrm{con}(X,\varrho)$ and hence *not* $(X,\varrho) \models \varphi$. According to 36.(i) this implies $(X,\varrho) \models \neg\varphi$ and hence $\neq \varphi \in \mathrm{con}(X,\varrho) \cap \mathrm{sen}(\mathcal{L}) = T$.

Now suppose $(X,\varrho) \models L$ even is a model of $L \subseteq \mathrm{form}(\mathcal{L})$. Now consider any formula $\varphi \in \mathrm{con}(L)$, that is $L \vdash \varphi$. By (iv) again, this implies $L \models \varphi$ and as $(X,\varrho) \models L$ together this implies $(X,\varrho) \models \varphi$. That is $\varphi \in \mathrm{con}(X,\varrho)$ and as φ has been arbitrary, this is

$$
\mathrm{con}(L) \quad \subseteq \quad \mathrm{con}(X,\varrho)
$$

□

8 Henkin Theory

Henkin Theory is concerned with assigning realizations to given theories. However explicit constructions of such models can be given in rare cases only, namely when the theory admits constant symbols to witness the truth of an existential statement (such theories will be called henkinian). Of course this requires the language to have sufficiently many constant symbols, and thus Henkin Theory is concerned with expanding languages and theories simultaneously until they become henkinian. This technique is employed in the proof of the Completeness Theorem in section 7. But before we can formally define henkinian theories and the henkinisation of theories we have to look at expansion of languages by adding constants in general:

Lemma 40: Constant Theorem:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$ be a many-sorted language and $\mathcal{D} = \{ d_j \mid j \in J \}$ be another set of constant symbols. We also assume, that the *sort* function $\mathrm{sort} : \mathcal{D} \to I$ has been enlarged to cover $\mathcal{D}$, as well. Then we regard

$$\mathcal{L}' \;:=\; \mathcal{L}_I\big(\mathcal{C} \cup \mathcal{D}, \mathcal{R}, \mathcal{F}, \mathrm{sort}\big)$$

an expansion of the language $\mathcal{L}$ by the constant symbols of $\mathcal{D}$. Now let $T \subseteq \mathrm{sen}(\mathcal{L})$ be a theory in $\mathcal{L}$ and to each $d_j \in \mathcal{D}$ assign a variable symbol y_j of $\mathcal{L}$ that does not appear in any of the sentences $\tau \in T$. That is we assume that $y : J \hookrightarrow \mathrm{var}(\mathcal{L}) : j \mapsto y_j$ is an injective map such that for any $j \in J$ we have $\mathrm{sort}(y_j) = \mathrm{sort}(d_j)$ and

$$\{ y_j \mid j \in J \} \cap \{ \tau[k] \mid \tau \in T, k \in 1 \dots \ell(\tau) \} \;=\; \emptyset$$

If $\varphi \in \mathrm{form}(\mathcal{L}')$ is a formula in the larger language, then we denote by $\varphi \in \mathrm{form}(\mathcal{L})$ the formula in the base language $\mathcal{L}$ that is gained by replacing any occurence of d_j by y_j. Formally that is: For any $k \in 1 \dots \ell(\varphi)$ we let

$$\varphi[k] \;:=\; \begin{cases} \varphi'[k] \text{ if } \varphi'[k] \notin \mathcal{D} \\ \quad y_j \text{ if } \varphi'[k] = d_j \end{cases}$$

(i) Then φ' is implied by T in the language of $\mathcal{L}'$ if and only if φ is implied by T in the language of $\mathcal{L}$. That is we find the equivalence of the following two statements

   (a) $T \models \varphi$ (in $\mathcal{L}$)

   (b) $T \models \varphi'$ (in $\mathcal{L}'$)

(ii) Then φ' can be deduced from T in the language of $\mathcal{L}'$ if and only if φ can be deduced from T in the language of $\mathcal{L}$. That is we find the equivalence of the following two statements

   (a) $T \vdash \varphi$ (in $\mathcal{L}$)

   (b) $T \vdash \varphi'$ (in $\mathcal{L}'$)

Proof:

(i) Note that explicitly (a) reads as: $\forall\,(X,\varrho) \in \mathrm{real}(\mathcal{L})$ and $\forall\,\omega \in \mathrm{ass}(X,\varrho)$ we have: If $\forall\,\tau \in T : \tau(\omega) = 1$ then $\varphi(\omega) = 1$, as well. Likewise (b) reads as: $\forall\,(X',\varrho') \in \mathrm{real}(\mathcal{L}')$ and $\forall\,\omega' \in \mathrm{ass}(X',\varrho')$ we have: If $\forall\,\tau \in T : \tau(\omega') = 1$ then $\varphi'(\omega') = 1$, as well.

So for the direction (a) $\implies$ (b) we are given some (X',ϱ') and some ω' such that for any $\tau \in T$ we have $\tau(\omega') = 1$ and we need to show $\varphi'(\omega') = 1$, too. Of course we fall back to (a) and define $X := X'$ and $\varrho := \varrho'$ restricted to $\mathcal{C}$. Also we let

$$\omega(x) \;:=\; \begin{cases} \omega'(x) \text{ if } x \notin y(J) \\ \varrho'(d_j) \text{ if } x = y_j \end{cases}$$

That is we transferred the information $\varrho'(\mathcal{D})$ to the unused variables $y(J)$. Then it is clear, that $\tau(\omega) = \tau(\omega') = 1$ and hence we have $\varphi'(\omega') = 1$ by assumption (a). But also $\varphi(\omega) = \varphi'(\omega') = 1$ we have what we needed to show.

The direction (b) $\implies$ (a) is just the reversal: we are given some (X,ϱ) and some ω such that for any $\tau \in T$ we have $\tau(\omega) = 1$ and we need to show $\varphi(\omega) = 1$. Again we fall back on (b), this time we define $X' := X$ and

$$\varrho'(e) \;:=\; \begin{cases} \varrho(e) \text{ if } e \in \mathcal{C} \\ \omega(y_j) \text{ if } e = d_j \end{cases}$$

Again we transferred the information $y(J)$, this time to ϱ'. Finally let $\omega' := \omega$, we no longer care for the y_j. Then it is clear, that $\tau(\omega') = \tau(\omega) = 1$ and hence $\varphi'(\omega') = 1$ by assumption. As $\varphi(\omega) = \varphi'(\omega') = 1$ this is what we wanted.

(ii) The direction (a) $\implies$ (b) is easy to see: Let us use induction on the number n of constants d_j that have been in φ' that is $\mathrm{free}(\varphi) \cap y(J) = \{\, y_{j_1}, \dots, y_{j_n} \,\}$. In case $n = 0$ we have $\varphi = \varphi'$ so $T \;\vdash\; \varphi$ readily implies $T \;\vdash\; \varphi'$, as $\mathcal{L} \subseteq \mathcal{L}'$. For $n = 1$ let y_j be any (free) variable of φ of the $\mathrm{sort}(y_j) = i$. As T is a theory (D5) gives us $T \;\vdash\; \forall_i y\,\varphi$. But as d_j is a constant we can also apply (L6) and modus ponens to find $T \;\vdash\; \varphi[y_j : d_j]$. This is a deduction in the language $\mathcal{L} + \{\, d_j \,\} \subseteq \mathcal{L}'$. For the induction step we note that $\varphi[y_j : d_j]$ has one less free variable of $y(J)$. By the induction hypothesis we have a deduction in $\mathcal{L} + \{\, d_{j_2}, \dots, d_{j_n} \,\} \subseteq \mathcal{L}'$:

$$T \;\vdash\; \psi \;:=\; \varphi[y_{j_n} : d_{j_n}] \dots [y_{j_2} : d_{j_2}]$$

Using the case $n = 1$ on ψ we find $T \;\vdash\; \psi[y_{j_1} : d_{j_1}] = \varphi'$ in $\mathcal{L}'$. Now for the direction (b) $\implies$ (a), that is we have a deduction $T \;\vdash\; \varphi'$ in $\mathcal{L}'$. That is there are finitely many formulae ψ'_1 to ψ'_m that are either logical axioms (L1) to (L7), (D3) or (D5) or formulae of T such that

$$\{\, \psi'_1, \dots, \psi'_m \,\} \;\vdash\; \varphi'$$

By renaming all variables $x_{i,j} \mapsto x_{i,2j}$ in all these formula we can assume that all these formulae only contain even-numbered variable symbols. Now let E be the set of all constant symbols in $\mathcal{D}$ appearing in any of the ψ'_r ($r \in 1 \dots m$) or in $\psi'_0 := \varphi'$

$$E \;:=\; y(J) \cap \{\, \psi'_r[k] \mid r \in 0 \dots m,\, k \in 1 \dots \ell(\psi'_r) \,\}$$

Clearly E is finite, so let us denote $E = \{ e_1, \dots, e_n \}$. Then for any $s \in 1 \dots n$ we find that $e_s = d_j$ for some $j \in J$. For this j the symbol $y_j = x_{i_s,p_s}$ is a variable symbol. Renaming $x_{i_s,p_s} \mapsto x_{i_s,2p_s+1}$ we can guarantee that all of the variable symbols used for any y_j are odd-numbered and therefore do not appear as variable symbols in any of the fromulae ψ'_1 to ψ'_m nor in φ'.

Let ψ_r be the formula of $\mathcal{L}$ obtained by substituting $d_j \in \mathcal{D}$ by $y_j \in \{ x_{i,2p+1} \in \mathrm{var}(\mathcal{L}) \mid i \in I, p \in \mathbb{N} \}$. By construction any ψ'_r is either gained by a logical axiom (L1) to (L7) or an application of a deduction rule (D1) to (D5). For (L1) to (L5), (L7) and (D1) to (D4) it is trivial, that this rule can also be applied to ψ_r in $\mathcal{L}$. (D5) has not been used on any of the d_j as these are constant symbols and not variable symbols. And the application on any variable symbol of $\mathcal{L}'$ is just the same in $\mathcal{L}$. In (L6) we have the restriction of x being freely substitutable by $t' \in \mathrm{term}(\mathcal{L}')$. But as the variable symbols of ψ_r and the y_j are disjoint $t \in \mathrm{term}(\mathcal{L})$ is freely substitutable again, so we can imitate this rule in $\mathcal{L}'$. Thus we have found a deduction

$$\{ \psi_1, \dots, \psi_m \} \ \vdash \ \varphi$$

in $\mathcal{L}$. And the ψ_r are either logical axioms or taken from $T \subseteq \mathrm{sen}(\mathcal{L})$ this truly is a deduction of the form $T \vdash \varphi$.

□

Definition 41:
Let $H \subseteq \mathrm{form}(\mathcal{L})$ be a set of formulae in the (many-sorted, first order) language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$, then H is said to be **henkinian**, iff for any formula $\varphi(x_{i,1}) \in \mathrm{form}(\mathcal{L})$ with the single free variable $\mathrm{free}(\varphi) = \{ x_{i,1} \}$ there is a constant symbol $c \in \mathrm{const}(\mathcal{L})$ of the $\mathrm{sort}(c) = i = \mathrm{sort}(x_{i,1})$ such that we can deduce

$$H \ \vdash \ \exists_i x_{i,1}\, \varphi \to \varphi[x_{i,1} : c]$$

Nota If H is henkinian and $\mathrm{free}(\varphi) = x_{i,1}$ and we can deduce $H \vdash \exists_i x_{i,1}\, \varphi$ then the modus ponens trivially yields $H \vdash \varphi[x_{i,1} : c]$. I.e. whenever H allows to prove the existence of an object, the language $\mathcal{L}$ already knows this object as a constant. Though henkinian theories have this property, the converse is not true - this property alone does not imply that a theory is henkinian.

Proposition 42:
Let $H \subseteq \mathrm{sen}(\mathcal{L})$ be a henkin theory in the (many-sorted, first order) language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$. Then the following statements are true:

(i) Let $s, t \in \mathrm{term}(\mathcal{L})$ be constant terms $\mathrm{free}(s) = \emptyset = \mathrm{free}(t)$ of $\mathcal{L}$ of the same $\mathrm{sort}(s) = i = \mathrm{sort}(t)$, then the following are equivalent

  (a) $H \vdash s =_i t$

  (b) $H \vdash \exists_i x\, (x =_i s \wedge x =_i t)$

  (c) There is a constant symbol $c \in \mathrm{const}(\mathcal{L})$ of $\mathrm{sort}(c) = i$, such that

$$H \ \vdash \ (c =_i s) \wedge (c =_i t)$$

(ii) Let R be a relation symbol or $\mathcal{L}$ of $\text{sort}(R) = (i_1, \ldots, i_n)$ and let $t_1, \ldots, t_n \in \text{term}(\mathcal{L})$ be constant terms $\text{free}(t_1) = \cdots = \text{free}(t_n) = \emptyset$ of $\text{sort}(t_k) = i_k$ (for $k \in 1 \ldots n$), then the following three statements are equivalent

(a) $H \vdash R(t_1, \ldots, t_n)$

(b) $H \vdash \exists_{i_1} x_1 \ldots \exists_{i_n} x_n \left(x_1 =_{i_1} t_1 \wedge \cdots \wedge x_n =_{i_n} t_n \wedge R(x_1, \ldots, x_n) \right)$

(c) There are constant symbols c_1, ..., $c_n \in \text{const}(\mathcal{L})$ of $\text{sort}(c_k) = i_k$, (for $k \in 1 \ldots n$) such that

$$H \vdash (c_1 =_{i_1} t_1) \wedge \cdots \wedge (c_n =_{i_n} t_n) \wedge R(c_1, \ldots, c_n)$$

Definition 43:
Let $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ be a formal (many-sorted, first order) language, $i \in I$ and $n \in \mathbb{N}$ then we recursively define new constant symbols and formal languages $\mathcal{L}_n$ according to the following scheme

$$\begin{aligned}
\mathcal{C}_{i,0} &:= \{ c \mid c \in \mathcal{C},\ \text{sort}(c) = i \} \\
\mathcal{C}_{i,1} &:= \{ c_\varphi \mid \varphi \in \text{form}(\mathcal{L}_0),\ \text{free}(\varphi) = x_{i,1} \} \\
\mathcal{C}_{i,n+1} &:= \{ c_\varphi \mid \varphi \in \text{form}(\mathcal{L}_n),\ \varphi \notin \text{form}(\mathcal{L}_{n-1}),\ \text{free}(\varphi) = x_{i,1} \} \\
\mathcal{L}_n &:= \mathcal{L}_I \left(\bigcup_{j=1}^{n} \bigcup_{i \in I} \mathcal{C}_{i,j},\ \mathcal{F},\ \mathcal{R},\ \text{sort} \right)
\end{aligned}$$

Thereby we assume that the new constant symbols c_φ of different formulae are different and are also different from the hereditary constant symbols of $\mathcal{C}$. The **henkinisation** of $\mathcal{L}$ is now defined to be the limit

$$\text{hen}(\mathcal{L}) := \mathcal{L}_I \left(\bigcup_{j \in \mathbb{N}} \bigcup_{i \in I} \mathcal{C}_{i,j},\ \mathcal{F},\ \mathcal{R},\ \text{sort} \right)$$

If now $L \subseteq \text{sen}(\mathcal{L})$ is a theory in $\mathcal{L}$ then we define its **henkinisation** to be the following (closed) theory

$$\begin{aligned}
H(L) &:= L \cup \left\{ \exists_i x_{i,1} \varphi \to \varphi[x_{i,1} : c_\varphi] \;\middle|\; \begin{array}{l} \varphi \in \text{form}(\text{hen}(\mathcal{L})), \\ \text{free}(\varphi) = \{ x_{i,1} \} \end{array} \right\} \\
\text{hen}(\mathcal{L}) &:= \overline{H(L)} = \{ \varphi \in \text{sen}(\text{hen}\mathcal{L}) \mid H(L) \vdash \varphi \}
\end{aligned}$$

Proposition 44:
If L, $K \subseteq \text{form}(\mathcal{L})$ are two sets of formulae in the many-sorted, formal language $\mathcal{L} = \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \text{sort})$ then the following statements are true:

(i) Any extension of a henkinian theory is henkinian itself, that is: If K is henkinian and $K \subseteq L$ then L is henkinian, as well.

(ii) The henkinisation of theories is order-preserving, that is if $K \subseteq L$ then $\text{hen}(K) \subseteq \text{hen}(L)$.

(iii) The extension $\text{hen}(L) : L$ of theories is conservative, that is we have the following property

$$\forall\, \varphi \in \text{form}(\mathcal{L}) \quad \text{we get} \quad L \vdash \varphi \iff \text{hen}(L) \vdash \varphi$$

(iv) If L is consistent, then $\mathrm{hen}(L)$ is a closed, consistent, henkinian theory in $\mathrm{hen}(\mathcal{L})$.

Proof of 42:

(i) Let $\varphi := (x =_i s) \wedge (x =_i t)$, then by 19.(vi) we have the tautology $\emptyset \vdash \varphi[x : s] \to \exists_i\, x\, \varphi$. For this particular φ we get

$$\emptyset \vdash (s =_i s) \wedge (s =_i t) \to \exists_i\, x\, \big((x =_i s) \wedge (x =_i t)\big)$$

Using 18 we can rewrite this as $\emptyset \vdash (s =_i s) \to ((s =_i t) \to \exists_i\, x\, \varphi$. But we also have $\emptyset \vdash s =_i s$ according to 19.(i) and hence by (D3)

$$\emptyset \vdash (s =_i t) \to \exists_i\, x\, \big((x =_i s) \wedge (x =_i t)\big)$$

Now for (a) $\Longrightarrow$ (b) we have $H \vdash s =_i t$ by assumption and hence modus ponens yields $H \vdash \exists_i\, x\, ((x =_i s) \wedge (x =_i t))$. For the next step (b) $\Longrightarrow$ (c) we use that H is henkinian: There is some constant symbol $c \in \mathcal{C}$ such that $H \vdash \exists_i\, x\, \varphi \to \varphi[x : c]$. But we have $H \vdash \exists_i\, x\, \varphi$ by assumption and therefore (by modus ponens again) $H \vdash (c =_i s) \wedge (c =_i t)$. Finally for (c) $\Longrightarrow$ (a) we use to turn $H \vdash (c =_i s) \wedge (c =_i t)$ into $H \vdash c =_i s$ and $H \vdash c =_i t$ by 17.(xii). Also $H \vdash c =_i s$ can be turned into $H \vdash s =_i c$ by 19.(ii). Finally we use 19.(ii) to generate $\emptyset \vdash (s =_i c) \to ((c =_i t) \to (s =_i t))$. This formula with $H \vdash s =_i c$ and $H \vdash c =_i t$ yields $H \vdash s =_i t$ by a double dose of modus ponens.

(ii) For (a) $\Longrightarrow$ (b) we note that $\emptyset \vdash t_k =_{i_k} t_k$ for any $k \in 1 \dots n$ by 19.(vi). Now by 18 this is $\emptyset \vdash (t_1 =_{i_1} t_1) \wedge \cdots \wedge (t_n =_{i_n} t_n)$. As also $H \vdash R(t_1, \dots, t_n)$ by (a) we get (for the same reason)

$$H \vdash \varphi_0 := \left(\bigwedge_{k=1}^{n} (t_k =_{i_k} t_k) \right) \wedge R(t_1, \dots, t_n)$$

Defining φ_1 to be the formula φ_0 where we replace t_n by x_n on the first and third (but not on the second) occasion, that is

$$\varphi_1 := \left(\bigwedge_{k=1}^{n-1} (t_k =_{i_k} t_k) \right) \wedge (x_n =_{i_n} t_n) \wedge R(t_1, \dots, t_{n-1}, x_n)$$

we have $H \vdash \varphi_0 = \varphi_1[x_n : t_n]$ and hence $H \vdash \exists_{i_n}\, x_n\, \varphi_1$ according to 19.(vi) and modus ponens. Next we continue with φ_2 which is defined by replacing t_{n-1} by x_{n-1} in $\exists_{i_n}\, x_n\, \varphi_1$ on the first and second occasion. The same reasoning yields $H \vdash \exists_{i_{n-1}}\, x_{n-1}\, \varphi_2$. Inductively we proceed until we arrive at the formula φ_n which is

$$\exists_{i_2}\, x_2 \dots \exists_{i_n}\, x_n\, (x_1 =_{i_1} t_1) \wedge \left(\bigwedge_{k=2}^{n} (x_k =_{i_k} t_k) \right) \wedge R(x_1, \dots, x_n)$$

Then $H \vdash \varphi_n[x_1 : t_1]$ implies $H \vdash \exists_{i_1}\, x_1\, \varphi_n$ and this finally is the formula given in (b).

Now for (b) $\Longrightarrow$ (c) we're going backwards in the above induction: By assumption we have $H \vdash \exists_{i_1}\, x_1\, \varphi_n$, as H is henkinian, there is a constant symbol $c_1 \in \mathcal{C}$ such that $H \vdash \varphi_n[x_n : c_n]$. Then we find c_2 and so on until we arrive at (c).

For (c) $\implies$ (a) we take to $\varphi := R(x_1, \ldots, x_n)$, then we have $H \vdash (c_1 =_{i_1} t_1) \wedge \cdots \wedge (c_n =_{i_n} t_n)$ and $H \vdash \varphi[x_1 : c_1, \ldots, x_n : c_n]$ by assumption and 18. But then we also get $H \vdash \varphi[x_1 : t_1, \ldots, x_n : t_n]$ by 19.(ix) and a double application of the modus ponens.

□

Proof of 44:

(i) By assumption we have $K \vdash \exists_i x_{i,1}\, \varphi \to \varphi[x_{i,1} : c]$ for any $\varphi(x_{i,1}) \in \mathrm{form}(\mathcal{L})$. Bus as $K \subseteq L$ we can use the dilution rule (D2) to also get $L \vdash \exists_i x_{i,1}\, \varphi \to \varphi[x_{i,1} : c]$.

(ii) By construction we have $H(K) \subseteq H(L)$ and thereby $\mathrm{con}(H(K)) \subseteq \mathrm{con}(H(L))$. And from this it is clear, that $\mathrm{hen}(K) \subseteq \mathrm{hen}(L)$, as well.

(iv) By definition of the henkinistaion it is clear, that $\mathrm{hen}(L)$ is a closed, henkinian theory. So it remains to show that $\mathrm{hen}(L)$ is consistent. Suppose $\mathrm{hen}(L)$ was contradictory, then there would be a formula $\gamma \in \mathrm{form}(\mathcal{L})$ such that $\mathrm{hen}(L) \vdash \gamma \wedge \neg\gamma$. By 38.(iv) $\mathrm{hen}(L) : H(L)$ is conservative and hence we obtain

$$H(L) \;\vdash\; \gamma \wedge \neg\gamma$$

Since we only allow finitely many steps for deducing, there are finitely many formulae $\vartheta_1, \ldots, \vartheta_m \in \mathrm{form}(\mathrm{hen}(\mathcal{L}))$, with $\mathrm{free}(\vartheta_j) = \left\{ x_{i_j,1} \right\}$, such that (denoting $x_j := x_{i_j,1}$ and $c_j := c_{\vartheta_j}$) we get

$$L \cup \left\{ \exists_{i_j} x_j\, \vartheta_j \to \vartheta_j[x_j : c_j] \mid j \in 1 \ldots m \right\} \;\vdash\; \gamma \wedge \neg\gamma$$

For any formula $\varphi \in \mathrm{hen}\mathcal{L}$ we now define its **level** to be the number of the iteration step such that φ first occurs as a formula of $\mathrm{hen}(\mathcal{L})$

$$\mathrm{level}(\varphi) \;:=\; \min\{\, l \in \mathbb{N} \mid \varphi \in \mathrm{form}(\mathcal{L}_l) \,\}$$

Without loss of generality we may enumerate the ϑ_j in such a way, that $k \leq l$ implies $\mathrm{level}(\vartheta_k) \leq \mathrm{level}(\vartheta_l)$. Now we denote

$$T_l \;:=\; T \cup \left\{ \exists_{i_j} x_j\, \vartheta_j \to \vartheta_j[x_j : c_j] \mid j \in 1 \ldots l \right\}$$

Claim $k < l \in 1 \ldots m \implies c_l$ does not occur in a formula of T_k

Proof Clearly c_l does not occur in any formula of T, since c_l does not belong to $\mathcal{L}$. Now suppose c_l would occur in one of the formulae $\vartheta_1, \ldots, \vartheta_k$, say in ϑ_j. Then, by definition of the henkinisation, ϑ_l would be a formula of $\mathcal{L}_n$ for $n = \mathrm{level}(\vartheta_j) - 1$. But $\vartheta_l \in \mathrm{form}(\mathcal{L}_n)$ would yield a contradiction

$$\mathrm{level}(\vartheta_l) \;\leq\; n \;<\; \mathrm{level}(\vartheta_j) \;\leq\; \mathrm{level}(\vartheta_l)$$

Now recall that (if T is contradictory) we found T_m to be such, too. Hence any formula may be deduced from it, in particular

$$T_m \;\vdash\; \neg\left(\exists_{i_m} x_m\, \vartheta_m \to \vartheta_m[x_m : c_m]\right)$$

Let us abbreviate $\gamma_m := \exists_{i_m} x_m\, \vartheta_m \to \vartheta_m[x_m : c_m]$ and note that $T_m = T_{m-1} \cup \{ \gamma_m \}$. But as the trivial rule (D1) also allows $T_{m-1} \cup \{ \neg\gamma_m \} \vdash \neg\gamma_m$ we already obtain by the section rule (D4)

$$T_{m-1} \;\vdash\; \neg\gamma_m \;=\; \neg\big(\exists_{i_m} x_m\, \vartheta_m \to \vartheta_m[x_m : c_m]\big)$$

By the definition of the conjunction we can reformulate this as $T_{m-1} \vdash \exists_{i_m} x_m\, \vartheta_m \wedge \neg\vartheta_m[x_m : c_m]$. And by 17.(x) this is again equivalent, to $T_{m-1} \vdash \exists_{i_m} x_m\, \vartheta_m$ and $T_{m-1} \vdash \neg\vartheta_m[x_m : c_m]$ Now we use another equivalence, the logical axiom (L5) to arrive at

$$T_{m-1} \vdash \neg\forall_{i_m} x_m \neg\vartheta_m \quad \text{and} \quad T_{m-1} \vdash \neg\vartheta_m[x_m : c_m]$$

Yet as c_m is a constant symbol that does not occur in any formula of T_{m-1} (by the above claim) it is as good as an unused variable symbol y of the same sort $\mathrm{sort}(y) = \mathrm{sort}(c_m) = i_m$. In other words we may apply the Constant Theorem to the sentence $\neg\vartheta_m[x_m : c_m]$, to get

$$T_{m-1} \vdash \neg\forall_{i_m} x_m \neg\vartheta_m \quad \text{and} \quad T_{m-1} \vdash \neg\vartheta_m[x_m : y]$$

Since all the formulae of T_{m-1} are sentences, we may apply the generalization rule (D5) to get

$$T_{m-1} \vdash \neg\forall_{i_m} x_m \neg\vartheta_m \quad \text{and} \quad T_{m-1} \vdash \forall_{i_m} y\, \neg\vartheta_m[x_m : y]$$

So we have proved, that T_{m-1} is inconsistent as well. Iterating the proof we see that $T_{m-2}, \dots, T_0$ all are inconsistent. But $T_0 = T$ was assumed to be consistent, a contradiction to the assumption, that $\mathrm{hen}(L)$ was contradictory.

(iii) The implication $L \vdash \varphi \implies \mathrm{hen}(L) \vdash \varphi$ is easy, as $L \subseteq H(L)$ we can apply (D2) to find $H(L) \vdash \varphi$. But as $\mathrm{hen}(L) = \overline{H(L)}$ we can invoke 38.(iv) to find $\mathrm{hen}(L) \vdash \varphi$.

In the converse direction we may assume that L is consistent (else $L \vdash \varphi$ would be true anyhow). Now suppose $\mathrm{hen}(L) \vdash \varphi$ but $L \not\vdash \varphi$, then $L' := L \cup \{ \neg\varphi \}$ is consistent, as well and $\mathrm{hen}(L) \vdash \varphi$ implies $\mathrm{hen}(L') \vdash \varphi$ by (ii) and (D2). But from (D1) we get $L' \vdash \neg\varphi$ and this implies $\mathrm{hen}(L') \vdash \neg\varphi$ as we have proved above. Yet by (vi) $\mathrm{hen}(L')$ is consistent, a contradiction to our assumption $L \not\vdash \varphi$.

□

Proposition 45: Henkin's Lemma
Let $H \subseteq \mathrm{form}(\mathcal{L})$ be a a theory, that is complete and henkinian, in the formal language $\mathcal{L} := \mathcal{L}_I(\mathcal{C}, \mathcal{F}, \mathcal{R}, \mathrm{sort})$. Then we obtain a well-defined realization (X, ϱ) of $\mathcal{L}$, by letting

- First define an equivalence relation $\sim$ on the set $\mathcal{C}$ of constant symbols of $\mathcal{L}$ by letting $c \sim d :\Longleftrightarrow \mathrm{sort}(c) = \mathrm{sort}(d)$ and $H \vdash c =_i d$. If we denote $\mathcal{C}_i := \{ c \in \mathcal{C} \mid \mathrm{sort}(c) = i \}$ (for $i \in I$) then the base sets X_i of the realization are given, by the constant symbols modulo this relation

$$X_i := \mathcal{C}_i \big/ \sim \;=\; \{ [c] \mid c \in \mathcal{C}, \mathrm{sort}(c) = i \}$$

- Next we define the interpretation ϱ: For a constant symbol $c \in \mathcal{C}$ we let $\varrho(c) := [c]$. For a relation symbol $R \in \mathcal{R}$ of $\text{sort}(R) = (i_1, \ldots, i_k)$ we define the relation $\varrho(R) \subseteq X_{i_1} \times \cdots \times X_{i_k}$, likewise for a function symbol $f \in \mathcal{F}$ of $\text{sort}(f) = (i_1, \ldots, i_k, i_{k+1})$ we get a well-defined function $\varrho(f) : X_{i_1} \times \cdots \times X_{i_k} \to X_{i_{k+1}}$ by letting

$$\varrho(R) := \{ ([c_1], \ldots, [c_k]) \in X_{i_1} \times \cdots \times X_{i_k} \mid H \vdash R(c_1, \ldots, c_k) \}$$

$$\varrho(f)([c_1], \ldots, [c_k]) := [c_{k+1}] \text{ where } H \vdash f(c_1, \ldots, c_k) = c_{k+1}$$

And this realization (X, ϱ) now has the following noteworthy properties

- For any term $t \in \text{term}(\mathcal{L})$ that is constant $\text{free}(t) = \emptyset$ and any constant symbol $c \in \mathcal{C}$ of the same $\text{sort}(c) = \text{sort}(t)$ we obtain the equivalence

$$H \vdash c =_i t \iff \varrho(t) = [c]$$

  Nota As t does not have free variables $t(\omega) = t(\omega')$ for any two assignments ω, $\omega' \in \text{ass}(X, \varrho)$. Hence we write $\varrho(t) := t(\omega)$ for an arbitrary assignment ω, as it does not matter anyhow.

- For any sentence $\varphi \in \text{sen}(\mathcal{L})$ we have $(X, \varrho) \models \varphi$ if and only if $H \vdash \varphi$. Nota In particular (X, ϱ) is a model of H, formally $(X, \varrho) \models H$.

Proof:
This proof of Henkin's Lemma will be conducted in several steps, first we will prove the well-definedness of the realization, which will already require the theory to be henkinian. Then we will proof the two claims in the proposition.

- Claim X is well-defined
  By 19.(i) to (iii) the relation $\sim$ clearly is an equivalence relation on $\mathcal{C}_i$ and hence every X_i is well-defined. We still need to show that all the X_i are non-empty: So for any $i \in I$ let $x := x_{i,1}$ and use 19.(iv) to pick up $\emptyset \vdash \exists_i x\, (x =_i x)$. Since H is henkinian we can use modus ponens to find a constant symbol c of $\text{sort}(c) = i$ such that $H \vdash c =_i c$. In particular $c \sim c$ and hence X_i is non-empty.

- Claim ϱ is well-defined
  The assignment $\varrho : c \mapsto \varrho(c) = [c]$ clearly is well-defined. For any relation symbol R we need to show, that $H \vdash R(c_1, \ldots, c_n)$ does not depend on the choice of the representants c_k. But if we have $c_k \sim d_k$ for any $k \in 1 \ldots n$ then $H \vdash c_k =_{i_k} d_k$ by definition of $\sim$. Hence by 17.(xii) we have $H \vdash (c_1 =_{i_1} d_1) \wedge \cdots \wedge (c_n =_{i_n} d_n)$. Combining this with the tautology 19.(ix) applied to $\varphi := Rx_1 \ldots x_n$ we find both $H \vdash Rc_1 \ldots c_n \to Rd_1 \ldots d_n$ and the other way around. Therefore, by definition of $\varrho(R)$ we find that $\varrho(R)([c_1], \ldots, [c_n])$ is equivalent, to $\varrho(R)([d_1], \ldots, [d_n])$.

  Similar to our handling of $\varrho(R)$ we find that $\varrho(f)$ is a well-defined partial function, due to the tautology 19.(vi): If $H \vdash f(c_1, \ldots, c_n) = c_{n+1}$ and $H \vdash f(d_1, \ldots, d_n) = d_{n+1}$ where $c_k \sim d_k$ then the tautology yields $H \vdash f(c_1, \ldots, c_n) = f(d_1, \ldots, d_n)$ and hence $H \vdash c_{n+1} =_{i_{n+1}} d_{n+1}$ according to 19.(iii). And this again is $[c_{n+1}] = [d_{n+1}]$.

It remains to check, that $\varrho(f)$ in fact is totally defined. Thus suppose we are given some constant symbols $c_1, \ldots, c_n$. Now apply the tautology 19.(v) to $t := fc_1 \ldots c_n$. For this t, this reads, as $\emptyset \vdash \exists_{i_{n+1}} x\,(x =_{i_{n+1}} f(c_1, \ldots, c_n))$. But since H is henkinian there is a constant c_{n+1} such that we may deduce $H \vdash c_{n+1} =_{i_{k+1}} f(c_1, \ldots, c_n)$ which by definition of $\varrho(f)$ yields $\varrho(f)([c_1], \ldots, [c_n]) = [c_{n+1}]$.

- Claim $H \vdash c =_i t \iff \varrho(t) = [c]$
  We will proof this statement by induction on the term calculus of $\mathcal{L}$ for t. So let us start with $t = d$ for some constant symbol d. Then we get the equivalences

$$H \vdash c =_i d \iff c \sim d \iff [c] = [d] = \varrho(t)$$

For the induction step we are given $t = f(s_1, \ldots, s_n)$. We now use the tautology 19.(v) for the s_k this time $\emptyset \vdash \exists_{i_k} x_k\,(x_k =_{i_k} s_k)$ for any $k \in 1 \ldots n$. Since H is henkinian we find constant symbols $c_1, \ldots, c_n$ of $\mathcal{L}$ with $H \vdash c_k =_{i_k} s_k$. But by the induction hypothesis this is equivalent to $\varrho(s_k) = [c_k]$. Now suppose $H \vdash c =_i f(s_1, \ldots, s_n)$ Then take a look at the tautology 19.(viii)

$$H \vdash \left(\bigwedge_{k=1}^{n} c_k =_{i_k} s_k \right) \to \big(f(c_1, \ldots, c_n) =_i f(s_1, \ldots, s_n)\big)$$

With modus ponens this becomes $H \vdash f(c_1, \ldots, c_n) =_i f(s_1, \ldots, s_k)$ and by 19.(iii) we find $H \vdash c =_i f(c_1, \ldots, c_k)$ from this. And by construction this is $[c] = \varrho(f)([c_1], \ldots, [c_n]) = \varrho(t)$. In the converse direction we suppose $\varrho(t) = [c]$, then we get $[c] = \varrho(t) = \varrho(f)([c_1], \ldots, [c_n])$ which by definition of $\varrho(f)$ yields $H \vdash c =_i f(c_1, \ldots, c_n)$. But since $H \vdash c_k =_{i_k} s_k$ the by the same arguments reversed, we find

$$H \vdash c =_i f(c_1, \ldots, c_k) =_i f(s_1, \ldots, s_k) =_i t$$

- Claim $(X, \varrho) \models \varphi \iff H \vdash \varphi$ for $\varphi = (s =_i t)$
  Since H is a henkinian theory we may apply 42.(i) in the first step of six equivalences. The rest is 17.(xii), the first claim that we just proved or purely by definition

$$\begin{aligned} H \vdash s =_i t &\iff \exists\, c \in \mathcal{C}_i : H \vdash (c =_i s) \wedge H \vdash (c =_i s) \\ &\iff \exists\, c \in \mathcal{C}_i : \big(H \vdash (c =_i s) \text{ and } H \vdash (c =_i t)\big) \\ &\iff \exists\, c \in \mathcal{C}_i : \big(\varrho(s) = [c] \text{ and } \varrho(t) = [c]\big) \\ &\iff \varrho(s) = \varrho(t) \\ &\iff (X, \varrho) \models (s =_i t) \end{aligned}$$

- Claim $(X, \varrho) \models \varphi \iff H \vdash \varphi$ for $\varphi = R(t_1, \ldots, t_k)$
  Since H is henkinian theory we may apply 42.(ii) in the first step of seven equivalences. The rest is 17.(xii), the first claim that we just proved or purely by definition

$$H \vdash \varphi \iff H \vdash R(t_1, \ldots, t_n)$$

$$\iff \begin{cases} \exists\, c_1, \ldots, c_n \in \mathcal{C} \\ H \vdash (\bigwedge_{k=1}^{n} (c_k =_{i_k} t_k)) \wedge R(c_1, \ldots, c_k) \end{cases}$$

$$\iff \begin{cases} \exists\, c_1, \ldots, c_n \in \mathcal{C} \\ H \vdash c_1 =_{i_1} t_1 \text{ and } \ldots \text{ and } H \vdash c_n =_{i_n} t_n \\ \text{and } H \vdash R(c_1, \ldots, c_n) \end{cases}$$

$$\iff \begin{cases} \exists\, c_1, \ldots, c_n \in \mathcal{C} \\ \varrho(t_1) = [c_1] \text{ and } \ldots \text{ and } \varrho(t_n) = [c_n] \\ \text{and } ([c_1], \ldots, [c_n]) \in \varrho(R) \end{cases}$$

$$\iff (\varrho(t_1), \ldots, \varrho(t_n)) \in \varrho(R)$$

$$\iff (X, \varrho) \models \varphi = R(t_1, \ldots, t_n)$$

- Claim $(X, \varrho) \models \varphi \iff H \vdash \varphi$ for $\varphi = \neg\varphi'$
For this step we note that $H \vdash \neg\varphi'$ is equivalent to *not* $H \vdash \varphi'$, since H is assumed to be a, complete theory. By induction hypothesis this now is *not* $(X, \varrho) \models \varphi'$. And using 36.(iii) this again is equivalent, to $(X, \varrho) \models \neg\varphi'$ which is $(X, \varrho) \models \varphi$.

- Claim $(X, \varrho) \models \varphi \iff H \vdash \varphi$ for $\varphi = \psi \to \chi$
Since φ is a statement, so are ψ and χ. Now, as H is complete $H \vdash \psi \to \chi$ is equivalent to "$H \vdash \psi$ implies $H \vdash \chi$". By induction hypothesis this again is equivalent, to "$(X, \varrho) \models \psi$ implies $(X, \varrho) \models \chi$. Now by 36.(iii) this is equivalent to $(X, \varrho) \models \psi \to \chi$ and we are done here.

- Claim $(X, \varrho) \models \varphi \iff H \vdash \varphi$ for $\varphi = \forall_i\, x\, \sigma$
Since - by assumption - φ is a statement σ may at most have the free variable x. In this step we assume that even φ is a statement, the case free$(\varphi) = \{ x \}$ will be dealt with in the subsequent two steps. In this case we can use the generalization rule (D5) to turn $H \vdash \sigma$ into $H \vdash \forall_i\, x\, \sigma$. Conversely if $H \vdash \forall_i\, x\, \sigma$, then we get $H \vdash \sigma$ from (L6). Altogether we have

$$H \vdash \sigma \iff H \vdash \forall_i\, x\, \sigma$$

On the other hand $(X, \varrho) \models \forall_i\, x\, \sigma$ is equivalent, to: $\forall\, \omega \in \text{ass}(X, \varrho)$ and $\forall\, a \in X_i$ we get $\sigma(\omega[x : a]) = 1$. But as free$(\sigma) = \emptyset$, $\omega[x : a]$ is irrelevant and hence this collapses to $\forall\, \omega \in \text{ass}(X, \varrho)$ we get $\sigma(\omega) = 1$. But this is $(X, \varrho) \models \sigma$. That is we also have the equivalence

$$(X, \varrho) \models \sigma \iff (X, \varrho) \models \forall_i\, x\, \sigma$$

Now by induction hypothesis we have $(X, \varrho) \models \sigma$ if and only if $H \vdash \sigma$. So combining all this we get the equivalence we sought

$$(X, \varrho) \models \varphi \iff (X, \varrho) \models \sigma \iff H \vdash \sigma \iff H \vdash \varphi$$

- Claim $H \vdash \varphi \implies (X, \varrho) \models \varphi$ for $\varphi = \forall_i\, x\, \varphi'(x)$
We need to show $(X, \varrho) \models \forall_i\, x\, \varphi'$, in other words this is $\forall\, \omega \in \text{ass}(X, \varrho)$ and $\forall\, a \in X_i$ we get $\varphi'(\omega[x : a]) = 1$. By construction $a = [c]$ for some $c \in \mathcal{C}_i$ and as φ' only has the free variable x the rest of ω is irrelevant. That is we have to prove

$$\forall\, c \in \mathcal{C}_i \;:\; \varphi'([c]) = 1$$

Since c is a constant symbol we have (L6) at hand, which provides the formula $\emptyset \;\vdash\; \forall_i\, x\, \varphi' \to \varphi'[x:c]$. By assumption we have $H \;\vdash\; \varphi$ and hence $H \;\vdash\; \varphi'[x:c]$ by modus ponens. Using the induction hypothesis of $\varphi'[x:c]$ we then find $(X,\varrho) \;\models\; \varphi'[x:c]$. That is we have found (for any ω) $1 = \varphi'[x:c](\omega) = \varphi'([c])$ as had been required.

- Claim $(X,\varrho) \models \varphi \quad\Longrightarrow\quad H \;\vdash\; \varphi \qquad$ for $\varphi = \forall_i\, x\, \varphi'(x)$
  Since H is henkinian there is a constant symbol c, such that we have the formula $H \;\vdash\; \exists_i\, x\, \neg\varphi' \to \neg\varphi'[x:c]$. But this is equivalent to $H \;\vdash\; \varphi[x:c] \to \neg\exists_i\, x\, \neg\varphi'$. By the logical axiom (L5) and 23.(ii) we can turn $\neg\exists_i\neg$ into $\forall_i$

$$H \;\vdash\; \varphi'[x:c] \to \forall_i\, x\, \varphi'$$

  By assumption we have $(X,\varrho) \models \forall_i\, x\, \varphi'(x)$ so in particular $(X,\varrho) \models \varphi'[x:c]$. Yet $\varphi'[x:c]$ is a statement (one step easier than φ') and hence we may use the induction hypothesis, obtaining $H \;\vdash\; \varphi[x:c]$. So if we apply modus ponens to the formula above, we arrive at $H \;\vdash\; \forall_i\, x\, \varphi'(x)$.

- Claim $(X,\varrho) \models \varphi \quad\Longleftrightarrow\quad H \;\vdash\; \varphi \qquad$ for $\varphi = \exists_i\, x\, \varphi'$
  By (L5) $H \;\vdash\; \exists_i\, x\, \varphi'$ is equivalent, to $H \;\vdash\; \neg\forall_i\, x\, \neg\varphi'$. And as H is complete this is equivalent to *not* $H \;\vdash\; \forall_i\, x\, \neg\varphi'$. As we have just seen, this is equivalent, to *not* $(X,\varrho) \;\models\; \forall_i\, x\, \varphi'$. As $\forall_i\, x\, \neg\varphi'$ is a sentence, this is again equivalent, to $(X,\varrho) \;\models\; \neg\forall_i\, x\, \neg\varphi'$. And for the realization (X,ϱ) it is clear that this now is equivalent, to $(X,\varrho) \;\models\; \exists_i\, x\, \varphi'$.

□

Remark 46: Conclusion:
In this article we formalized deductions $T \;\vdash\; \varphi$ from a given set of statements T to a new result φ (sections 2 and 4) and provided lots of tautologies to get a feel for this method and to establish the standard arguments used in mathematical proofs. The most noteworthy result (from the point of view of naive mathematics) are the commutativity rules 24.(i) to (iii) for quantifiers.
Then we introduced $T \;\models\; \varphi$ (in section 6) that is a formal way of looking at the mathematical practice how proofs are conducted. In the end we used Henkin's Theory (in this section 8) to prove the equivalence of $T \;\vdash\; \varphi$ and $T \;\models\; \varphi$ (in section 7). This has been the aim of this article, right from the start. And it is a much more important result, than Gödel's Incompleteness Theorem that came to the attention of non-mathematicians, too.
But whenever one story ends, the next story begins. The Compactness Theorem - readily apparent for $T \;\vdash\; \varphi$, but astounding for $T \;\models\; \varphi$ - gives rise to new theorems, like the up- and down-theorems of Löwenheim-Skolem. And the methods we displayed here are the founding stone of whole new fields of mathematics: (Algebraic) Model Theory up to the method of "forcing". We displayed a few exemplary results, that can be gained by these methods, in a showcase in section 3, but without proof. These are stories for another day.

9 Open Publication License

I. Copyright

The copyright to each Open Publication is owned by its author(s) or designee.

II. Scope of License

The following license terms apply to all Open Publication works, unless otherwise explicitly stated in the document. Mere aggregation of Open Publication works or a portion of an Open Publication work with other works or programs on the same media shall not cause this license to apply to those other works. The aggregate work shall contain a notice specifying the inclusion of the Open Publication material and appropriate copyright notice.

Severability: If any part of this license is found to be unenforceable in any jurisdiction, the remaining portions of the license remain in force.

No Warranty: Open Publication works are licensed and provided "as is" without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement.

III. Requirements on both unmodified and modified Versions

Any publication in standard (paper) book form shall require the citation of the original publisher and author. The publisher and author's names shall appear on all outer surfaces of the book. On all outer surfaces of the book the original publisher's name shall be as large as the title of the work and cited as possessive with respect to the title.

IV. Requirements on Modified Works

All modified versions of documents covered by this license, including translations, anthologies, compilations and partial documents, must meet the following requirements:

- The modified version must be labeled as such.
- The person making the modifications must be identified and the modifications dated.
- Acknowledgement of the original author and publisher if applicable must be retained according to normal academic citation practices. The location of the original unmodified document must be identified.
- The original author's (or authors') name(s) may not be used to assert or imply endorsement of the resulting document without the original author's (or authors') permission.

V. Good-Practice Recommendations

In addition to the requirements of this license, it is requested from and strongly recommended of redistributors that:

- If you are distributing Open Publication works on hardcopy or CD-ROM, you provide e-mail notification to the authors of your intent to redistribute at least thirty days before your manuscript or media freeze, to give the authors time to provide updated documents. This notification should describe modifications, if any, made to the document.
- All substantive modifications (including deletions) be either clearly marked up in the document or else described in an attachment to the document. Finally, while it is not mandatory under this license, it is considered good form to offer a free copy of any hardcopy and CD-ROM expression of an Open Publication-licensed work to its author(s).

VI. License Options

Distribution of the work or derivative of the work for commercial purposes is prohibited, unless prior permission is obtained from the copyright holder, in written form.

If you have read this text I would like to invite you to contribute to it:
Comments, corrections and suggestions are very much appreciated, at

zdl@pmhs.de, and visit my homepage at

https://e.pcloud.link/publink/show?code=kZBvhnZEqstjOJc17uS3meyVfAoOSLhRkLk

This text is dedicated to the entire mathematical society:

To all those who contribute to mathematics

by research or publication or keep it alive by teaching it.

References

[1] John L. Bell and Moshe Machover, *A Course in Mathematical Logic*, North Holland Publishing Company, Amsterdam, 1977.

[2] Herbert B. Enderton, *A mathematical Introduction to Logic (2nd edition)*, Academic Press, London, 2001. Available at https://archive.org/details/mathematical-introduction-to-logic-herbe/mode/2up.

[3] Ulrich Felgner, *Modelltheoretische Algebra (lecture notes)*, Eberhard Karls Universität Tübingen, Tübingen, 1993.

[4] Kurt Gödel, *Über die Vollständigkeit des Logikkalküls* (1928), 136–140. Available at https://www.researchgate.net/publication/380197149_Godels_Doktorarbeit_1928-30_Die_Vollstandigkeit_der_Axiome_des_logischen_Funktionenkalkuls.

[5] Leon Henkin, *The Completeness of the First-Order Functional Calculus*, Journal of Symbolic Logic **14** (1949), 159-166.

[6] Georg Kreisel and Jean-Louis Krivine, *Model Theory*, Springer-Verlag, Berlin, 1972.

[7] Patrick Morandi, *Field and Galois Theory*, Springer Verlag, Graduate Texts in Mathematics 167, New York, 1996.

[8] Mojżesz Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welcher die Addition als einzige Operation hervortritt*, Comptes-rendus du premiére Congrés des Mathémaciens des Pays Slaves (1929), 92-101.

[9] A. Bernhard Zeidler, *Abstract and Commutative Algebra*, 2025. Available at https://www.academia.edu/120645082/Abstract_and_Commutative_Algebra.